PATENT ABSTRACTS OF JAPAN

(11)Publication number : **2001-290700**
(43)Date of publication of application : **19.10.2001**

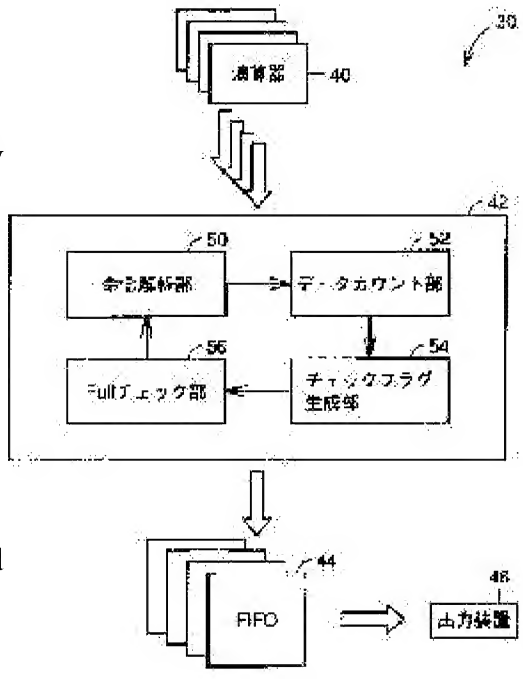
(51)Int.Cl. G06F 12/02
G06F 5/06
G06F 12/00
G06F 13/38

(21)Application number : **2000-106780**
(22)Date of filing : **07.04.2000**

(71)Applicant : **MITSUBISHI ELECTRIC CORP**
(72)Inventor : **OBARA JUNKO**
KAWAI HIROYUKI
INOUE YOSHITSUGU
STREITENBERGER ROBERT

(54) OUTPUT FIFO DATA TRANSFER CONTROLLER

(57)Abstract:
PROBLEM TO BE SOLVED: To provide an output FIFO data transfer controller, capable of accelerating a data transfer processing speed, while efficiently utilizing FIFO areas by unnecessitating the confirmation of empty areas in a FIFO by means of a program.
SOLUTION: An output FIFO data transfer controller 42 is provided with an instruction analyzing part 50 for calculating the amount of data to be transferred by analyzing the instruction of data transfer to an output FIFO storage device 44 provided with plural banks, data count parts 52 and 54 for calculating the amount of data written in the banks under outputting from the data amount calculated by the instruction analyzing part 50 and outputting a decision flag showing whether the free capacity of the bank under outputting satisfies prescribed conditions or not and a full check part 56 for prohibiting the processing of the next instruction, until resetting the decision flag from the data count parts 52 and 54 or full flag outputted from the output FIFO storage device 44.



CLAIMS

[Claim(s)]
[Claim 1]An output FIFO data transfer controller comprising:
Consist of two or more banks and a data transfer instruction by a computing unit to an output FIFO storage which has a function which outputs a full flag when a bank next to a bank under writing is a full state is analyzed, While computing data volume transmitted, from data volume computed by a command analysis circuit written in said output FIFO storage, and said command analysis circuit, the data concerned. Data count circuits which compute data volume written in a bank under output, judge whether an availability of a bank under said output satisfies predetermined conditions, and output a decision flag.
A full checking circuit which forbids processing of the next command until said full flag outputted from a decision flag or said output FIFO storage from said data count circuits is reset.

[Claim 2]Said data count circuits from data volume computed by said command analysis circuit. The output FIFO data transfer controller according to claim 1 which computes data volume written in a bank under output, judges whether conditions of being more than the maximum data volume which an availability of a bank under said output can transmit at once are satisfied, and outputs a decision flag.
[Claim 3]An output FIFO data transfer controller comprising:
A command analysis circuit which writes the data concerned in said output FIFO storage while computing data volume transmitted by analyzing a data transfer instruction by a computing unit to an output FIFO storage which outputs a read signal which shows the number of data read when data was read.
Data volume computed by said command analysis circuit.
Data count circuits which compute an availability of said output FIFO storage, judge whether an availability of said output FIFO storage satisfies predetermined conditions, and output a decision flag from said read signal.
A full checking circuit which forbids processing of the next command until a decision flag from said data count circuits is reset.

[Claim 4]The output FIFO data transfer controller according to claim 3 with which said output FIFO storage includes two or more banks.
[Claim 5]The output FIFO data transfer controller according to claim 3 including a bank with said single output FIFO storage.
[Claim 6]Data volume with which said data count circuits were computed by said command

analysis circuit, An availability of said output FIFO storage is computed from said read signal, The output FIFO data transfer controller according to any one of claims 3 to 5 which judges whether conditions of being more than the maximum data volume which an availability of said output FIFO storage can transmit at once are satisfied, and outputs a decision flag.

DETAILED DESCRIPTION

[Detailed Description of the Invention]
[0001]
[Field of the Invention]In a processor with two or more computing units, this invention the transmission at the time of transmitting arbitrary numbers of data to an output unit via a pushup storage (only referred to as "FIFO" below.) with one command about the output FIFO data transfer controller for controlling especially, It is related with the output FIFO data transfer controller which can prevent overflow of FIFO.
[0002]
[Description of the Prior Art]In the processor etc. which have two or more computing units, in order to absorb the difference of processing timing with an output unit, FIFO is used. The processor 140 concerning a Prior art with reference to drawing 22, The output of two or more computing units 40, the output FIFO data transfer controller 142 which undergoes the output of the computing unit 40, and the output FIFO data transfer controller 142 is held temporarily, and output FIFO44 for outputting to the output unit 46 is included.
[0003]The instruction analyzing section 150 for the output FIFO data transfer controller 142 to analyze the write instruction of output FIFO44, The data counting part 152 for counting the number of the data written in output FIFO44, The check flag generation part 154 for generating the flag for investigating whether output FIFO44 is Full and the Full check part 156 for judging whether output FIFO44 is in a Full state based on the Full flag which the check flag generation part 154 generated are included.
[0004]With reference to drawing 23, this output FIFO data transfer controller 142 operates as follows. First, the instruction analyzing section 150 analyzes the write-in command to output FIFO44 from the computing unit 40, and writes data in output FIFO44 (Step 1.). By the following explanation, a "step" is only indicated to be "S". . The data counting part 152 counts the data number written in output FIFO44 by what (S2) a count **********ed for whenever data is written in output FIFO44 by the instruction analyzing section 150. The check flag generation part 154 sets a check flag (fullcheck), when the value of the count of the data counting part 152 is more than the size of output FIFO44.
[0005]On the other hand, in output FIFO44, when the writing to the bank next to the bank under present writing is not made, by setting flag OFIFO_full_flag shows it. The Full check part 156 takes the logical product of OFIFO_full_flag and fullcheck, when the result is 1, judges with the writing of output FIFO44 being impossible, and sets the value of the flag fullcheck as 1 (S3). therefore, the Full check part 156 stops the writing of output FIFO44 by the next command until all the data of the following bank of output FIFO44 is read in this case, the following bank serves as empty, and OFIFO_full_flag is set to zero namely, -- (S5).
[0006]
[Problem(s) to be Solved by the Invention]In this conventional method, after data having been written in the end of the bank which has output FIFO44, for example, and having reached at the end of the bank is detected, it is judged whether data can be written in the following bank. That

is, after data is written in in the end of a bank, it is judged whether the writing of the data to the following bank is possible.
[0007]Therefore, if the data transfer which must be stored ranging over two or more banks with one command arises in the conventional method, Also when the judgment that the writing to the following bank is impossible was made by the Full check part 156, there was a problem that the writing could not be stopped.
[0008]In order to solve such a problem, before performing processing which writes data in output FIFO44 by the program side which a processor executes, conventionally, the Full state of output FIFO44 always needed to be checked. For this check processing, when the conventional output FIFO data transfer controller is used, the fall of the data-processing transfer rate is caused.
[0009]For example, with reference to drawing 24, it is assumed that output FIFO44 is 4 bank configurations. The end of the bank 2 has an opening for one data so that it may be illustrated, but the bank 3 shall be write-in ending altogether. At this time, by the conventional method, when the data of the last of the bank 2 is written in, the FULL check is performed. Therefore, when it is the command that the command which writes in the data of the last of the bank 2, for example transmits two or more data (for example, three data), in spite of not having read the data for two pieces of the bank 3 yet, it will be overwritten.
[0010]In order to keep such a problem from arising, it is necessary to confirm before transmission whether output FIFO44 is Full by the program side as mentioned above. A case may need to perform this check at every command issue each time.
[0011]One proposal for solving such a problem is indicated by JP,11-161467,A. In the art of an indication in this gazette, a memory is divided into two blocks on a certain boundary, and each is used as another FIFO. And while having prevented the circuitry of writing / read-out control circuit from devising the calculation (determination of next address) method of the next write-in position of each FIFO, and becoming complicated, It is judged whether free space is insufficient, respectively and whether the size and free space of data which are written in the next are compared for every FIFO, and it can write in.
[0012]However, in this art, since the number of FIFO is limited to two, mass data cannot be carried. Since two blocks are used as separate FIFO, there is a problem that the utilization ratio of a memory area is bad.
[0013]JP,63-167949,A is indicating the data transfer system with the sufficient efficiency which comprises two or more FIFO connected in series. In this art, it had the information which shows whether a FIFO buffer is empty for every FIFO, and grasps how much free space there is as the whole FIFO by it. And the size of free space and the size of data transmitting which were obtained in this way are compared, and only when the free space is larger, data is written in the FIFO buffer.
[0014]However, in this art, if it has the information which shows whether it is empty for every FIFO and at least one data is written in one FIFO, it will be supposed that it is not that FIFO empty. Therefore, the case where the field cannot be used also produces FIFO which still has free space actually. Therefore, the whole field of FIFO cannot be used efficiently too, but there is a problem that transfer efficiency also falls as a result.
[0015]so, the free space of FIFO according [the purpose of this invention] to a program -- it is providing the output FIFO data transfer controller which can raise data transfer processing speed by supposing that it is unnecessary and using a FIFO area efficiently.
[0016]
[Means for Solving the Problem]An output FIFO data transfer controller concerning this

invention includes a command analysis circuit which writes the data concerned in an output FIFO storage while computing data volume transmitted by analyzing a data transfer instruction by a computing unit to an output FIFO storage which consists of two or more banks. An output FIFO storage has a function which outputs a full flag, when a bank next to a bank under writing is a full state. Further this output FIFO data transfer controller from data volume computed by command analysis circuit. Data count circuits which compute data volume written in a bank under output, judge whether an availability of a bank under output satisfies predetermined conditions, and output a decision flag, A full checking circuit which forbids processing of the next command is included until a full flag outputted from a decision flag or an output FIFO storage from data count circuits is reset.

[0017]An availability of a bank under output of an output FIFO storage is calculated after instruction processing, conditions with the availability are not satisfied, and when the following bank is a full state, processing of the next command is forbidden. Since it is judged whether there is any inconvenience although the following data is written in before reaching to the last of a bank, there is no possibility that overwrite of data may arise. It is not necessary to judge whether it is full about each of two or more banks at this time. It is not necessary to perform processing which prevents overwrite of data by a program.

[0018]Preferably, data count circuits compute data volume written in a bank under output from data volume computed by command analysis circuit, judge whether conditions of being more than the maximum data volume which an availability of a bank under output can transmit at once are satisfied, and output a decision flag.

[0019]Since processing of the next command is started when an availability more than the maximum data volume which can be transmitted at once is secured, or when the following bank is no longer full, there is no possibility that it may be overwritten even if data of a peak is transmitted to an output FIFO storage with the next command.

[0020]According to other aspects of affairs of this invention, an output FIFO data transfer controller includes a command analysis circuit which writes the data concerned in an output FIFO storage while computing data volume transmitted by analyzing a data transfer instruction by a computing unit to an output FIFO storage. An output FIFO storage outputs a read signal which shows the number of data read when data was read from an output FIFO storage. Data volume with which this output FIFO data transfer controller was further computed by command analysis circuit, Data count circuits which compute an availability of an output FIFO storage, judge whether an availability of an output FIFO storage satisfies predetermined conditions, and output a decision flag from a read signal, A full checking circuit which forbids processing of the next command is included until a decision flag from data count circuits is reset.

[0021]An availability of an output FIFO storage is calculated after instruction processing, and when conditions with the availability are not satisfied, processing of the next command is forbidden. Since the next command is processed after it is checked that there is certainly sufficient availability, there is no possibility that overwrite of data may arise. It is not necessary to perform processing which prevents overwrite of data by a program, and improvement in the speed of processing can be attained.

[0022]An output FIFO storage may also include two or more banks, and may also include only a single bank. In any case, transmission processing of data is accelerable.

[0023]Data volume with which data count circuits were preferably computed by command analysis circuit, From a read signal, an availability of an output FIFO storage is computed, it judges whether conditions of being more than the maximum data volume which an availability of

an output FIFO storage can transmit at once are satisfied, and a decision flag is outputted.
[0024]The next command is processed only at the time more than the maximum data volume which an availability of an output FIFO storage can transmit at once. Therefore, even if data volume transmitted with the next command is the maximum data volume, there is no possibility that overwrite of data may arise. Therefore, it is unnecessary to perform processing for avoiding overwrite of data in a program, and it can accelerate processing.
[0025]
[Embodiment of the Invention]With reference to embodiment 1 drawing 1, the processor 30 containing the output FIFO data transfer controller concerning the embodiment of the invention 1 is provided with the following.
Two or more computing units 40.
The output FIFO data transfer controller 42 which receives the data in which this computing unit 40 is outputted.
Output FIFO44 which consists of two or more banks for being controlled by the output FIFO data transfer controller 42, holding the data from the computing unit 40 temporarily, and transmitting to the output unit 46.

[0026]The output FIFO data transfer controller 42 is provided with the following.
The instruction analyzing section 50 which computes the data number (m) which analyzes the write-in command to output FIFO44, and is outputted to output FIFO44 with the command concerned.
The data counting part 52 for counting the data currently written in each bank of output FIFO44.
The check flag generation part 54 which generates the flag for investigating whether the bank of output FIFO44 is in a Full state.
The Full check part 56 for investigating whether the size of the free space of output FIFO44 has satisfied predetermined conditions, and forbidding the processing to the next command by the instruction analyzing section 50 according to the result.

[0027]With reference to drawing 2, this output FIFO data transfer controller 42 operates as follows. First, the instruction analyzing section 50 computes the number of the data which analyzes a command and then should be written in output FIFO44 (m) (S11), and writes this data in output FIFO44. The data counting part 52 adds the value of m to the value of the counter in which the number of the data currently written in each bank of output FIFO44 is shown. That is, the value Counter of a counter is calculated by the following formulas (S21).
[0028]Counter = Counter+m -- (1) check-flag generation part 54 compares the number (NN=N-M) which subtracted from the value Counter of a counter, and the size (N) of output FIFO44 the maximum number (M) of the data which can be written in with one command. When the value Counter of a counter is more than NN, the check flag generation part 54 sets the check flag fullcheck, and, in other cases, resets it (S13). If this is written in accordance with the notation of the C language, it will become as follows.
[0029]
fullcheck = (NN=(N-M))<=counter) ? 1:0 -- the (2) Full check part 56, A logical product with the check flag fullcheck which flag OFIFO_full_flag (set to 0 when other [it cannot do, and], 1 and) which shows whether the writing to the bank next to output FIFO44 is made, and the check flag generation part 54 generated is taken (S14). And as a result, it is judged whether OFIFO_full is 1 (S15). When result OFIFO_full is 1, processing returns to S14 (it is "YES" at S15). As a

result, processing of S14 and S15 is repeated until the data of output FIFO44 is read and OFIFO_full_flag is reset, and processing of the command which writes in the next data of output FIFO44 is stopped. As for OFIFO_full_flag, N-1 in the bank of N FIFO shall be set, when data was written in, all the data is in the state which has not been read yet and data is written in the Nth bank for the first time.
[0030]If the result of a judgment of S15 serves as "NO", processing will return to S11, and processing to the next command is performed.
[0031]By having such composition, the number of the data written in output FIFO44 counts, before writing data in output FIFO44, and it is investigated in the end of a bank whether the size of the free space of output FIFO44 has always satisfied predetermined conditions. When data transfer which straddles the boundary of a bank occurs, it becomes unnecessary as a result, to be able to judge whether there is any possibility that a bank may serve as Full at high speed by hardware, and to judge whether a bank is Full by the program side. Software processing with slow processing speed is avoided, and a data transfer rate can be raised.
[0032]A concrete example is given and operation of the output FIFO data transfer controller of this Embodiment 1 is explained.
[0033]As shown in drawing 3, output FIFO44 shall be [the size of each bank] 32 including four banks Bank 1-4. At this time, the value of NN=N-M shown in a formula (2) is NN=32-4=28. Therefore, when it comes to Counter>=28, fullcheck is set.
[0034]Data shall be finishing (unread appearance) in writing at the portion shown with the slash in drawing 3. That is, the banks 1, 3, and 4 shall be write-in ending altogether, only the bank 2 shall write them in by 25 pieces, they shall be ending, and shall be unread appearance. The number of computing units shall be four and a maximum of four data shall be simultaneously transmitted with one command. The bank under writing is the bank 2. Since the writing to the following bank (bank 3) is not made at this time, OFIFO_full_flag is set.
[0035]In this state, the following commands shall be published in order and the state of output FIFO44 at that time is explained.
[0036]
Data transfer (1) (S11: drawing 2) of C:4 data transfer instructions of B:3 data transfer instructions of A:1 command
The Full check part 56 analyzes the command A, and m= 1 is calculated. One data is written in output FIFO44 (drawing 4).
[0037](2)(S12)
Counter=Counter+1 is calculated in the data counting part 52. Therefore, Counter=25+1=26 is calculated.
[0038](3)(S13)
In the check flag generation part 54, since Countet>=28 is not realized, fullcheck is not set.
[0039](4) (S14, 15)
Since fullcheck is 0, processing returns to S11 as a result of the judgment in the check flag generation part 54.
[0040](5)(S11)
In the instruction analyzing section 50, the next command (command B) is analyzed and m= 3 is calculated. Furthermore, three data is written in to output FIFO44 (drawing 5).
[0041](6)(S12)
Counter=26+3=29 is calculated in the data counting part 52.
[0042](7)(S13)

In the check flag generation part 54, since Counter>=28 is realized, fullcheck is set.
[0043](8)(S14)
In the Full check part 56, since fullcheck is 1 also in 1 and OFIFO_full_flag, OFIFO_full is set to one.
[0044](9)(S15)
processing (S14 and S15) of the Full check part 56 is repeated [since OFIFO_full is 1,] until OFIFO_full_flag is reset, and the data of the bank 3 of output FIFO44 is read namely,. The data of the bank 3 is read, and if OFIFO_full_flag is reset, processing will return to S11 as a result of the judgment by the Full check part 56. The state of output FIFO44 at this time is shown in drawing 6.
[0045](10)(S11)
The instruction analyzing section 50 analyzes the next command C, calculates m= 4, and writes in four data to output FIFO44.
[0046](11)(S12)
Counter=29+4=33 is calculated in the data counting part 52.
[0047](12)(S13)
Since Counter>=28 is realized, the check flag generation part 54 sets fullcheck.
[0048](13)(S14)
OFIFO_full_flag is 0. Therefore, OFIFO_full which is a decision result of the Full check part 56 is also set to zero.
[0049](14)(S15)
Since OFIFO_full is 0, it returns to S11 (drawing 7). When data transfer is performed by next command, the data of the bank 3 is already read-out settled. Therefore, there is no possibility that overwrite of data may arise.
[0050]As mentioned above, according to the output FIFO data transfer controller 42 of this embodiment, when writing data in output FIFO44, it asks for that data number and the size of the free space of that bank was investigated in the end of a bank. And when a bank may be straddled and data may be written in, the next command is not received until it will be in the state where data transfer to output FIFO44 can be performed certainly. Therefore, even if the writing of data which straddles generates a bank, overflow can be avoided, and there is no possibility that it may be overwritten by the data which is not read. Therefore, it is not necessary to take the policy for preventing overwrite of such data by the program side. As a result, improvement in data transfer processing speed can be aimed at.
[0051]At Embodiment 1 described on Embodiment 2, it can transmit well with the composition which output FIFO44 consists of two or more banks, therefore was described above. However, OFIFO_full_flag will not be reset, if data is also partly written in even if it is a case where this bank has free space actually when only one piece has a bank and output FIFO tends to apply the same method as Embodiment 1. In the meantime, the writing of data stops. Therefore, there is a possibility that data transfer processing speed may fall.
[0052]The output FIFO data transfer controller explained in this Embodiment 2 is improved so that data transfer may be performed at high speed, even when the bank of output FIFO is one. Therefore, according to this Embodiment 2, it investigates how much free space not only the flag that shows [in which FIFO will only make it empty completely] whether it comes out but FIFO has, when there is free space, OFIFO_full_flag is reset, and when there is nothing, it sets. The output FIFO data transfer controller applied to this Embodiment 2 below is explained in detail.
[0053]With reference to drawing 8, the processor 70 containing the output FIFO data transfer

controller 72 concerning this Embodiment 2 is provided with the following.
Computing unit 40.
Output FIFO data transfer controller 72.
Output FIFO74 for consisting of one bank and outputting the data from the output FIFO data transfer controller 72 to the output unit 46.

[0054]The output FIFO data transfer controller 72 is provided with the following.
The instruction analyzing section 80 for analyzing the write instruction of output FIFO74.
The data counting part 82 for investigating whether output FIFO74 has free space.
The Full check part 86 for investigating whether output FIFO74 is in a Full state.

[0055]With reference to drawing 9, the output FIFO data transfer controller 72 operates as follows. If the write instruction of the data from the computing unit 40 to output FIFO74 is received, the instruction analyzing section 80 will ask for the data number (m) written in, and will write data in output FIFO74 (S21). The data counting part 82 adds the value m which the instruction analyzing section 80 calculated to the value num of the counter showing the data number stored in output FIFO74 which he is maintaining. That is, the following calculations are performed (S22).
[0056]num=num+m Further, --(4) data counting part 82 subtracts only the read number (referred to as r.) from the value num of a counter, when data is read from output FIFO74. That is, the following calculations are performed (S23).
[0057]num=num-r -- (5) data counting part 82 judges whether it is more than the value (N-M) in which the value of the value num of the counter produced by calculating in this way subtracted from the size (N) of output FIFO74 the maximum number (M) of the data which can be transmitted with one command (S24). If it is num>N-M, flag OFIFO_full_flag_m will be set, and OFIFO_full_flag_m will be reset if it is except it (S24). If this is expressed in accordance with the notation of the C language, it will become as follows.
[0058]
OFIFO_full_flag_m=(num>=N-M) ? 1:0 -- (6) and the Full check part 86 repeat processing of S23-S25 until it will return to S23 (S25) and OFIFO_full_flag_m will be reset, if OFIFO_full_flag_m is 1. That is, the output FIFO data transfer controller 72 does not receive the next command until data is read from output FIFO74 and the conditional expression in a formula (6) stops concluding. If the conditional expression in a formula (6) is concluded, control will return to S21 and will process the next command.
[0059]A concrete example explains operation of this output FIFO data transfer controller 72. As shown in drawing 10, size shall have one bank by 32 output FIFO74. As for the number of the computing units 40, four, therefore a maximum of four data shall be transmitted simultaneously. NN=N-M of a formula (6) is set to NN=32-4=8 at this time. Namely, the output FIFO data transfer controller 72 sets OFIFO_full_flag_m at the time of num>=28.
[0060]For example, as shown in drawing 10, 25 data is written in output FIFO74, and the case where one piece is not read, either is assumed (num=25). At this time, the following commands shall be published and two data shall be read between the command A and the command B.
[0061]
A:1 command transmission (two-piece read-out)
According to concrete operation, the state of output FIFO74 changes as follows at the time of C:4 B:3 command transfer command ******.

[0062](1) (S21)
The command A is analyzed in the instruction analyzing section 80, and m= 1 is calculated. One data is written in output FIFO74.
[0063](2)(S22)
num=25+1=26 is calculated in the data counting part 82 (drawing 11).
[0064](3)(S23)
It answers that two data was read and num=26-2=24 is calculated in the data counting part 82 (drawing 12).
[0065](4)(S24)
Since num>=28 is not satisfied, the Full check part 86 resets OFIFO_full_flag_m.
[0066](5)(S25)
Since OFIFO_full_flag_m is 0, control returns to S21 and processing of the next command (command B) is started.
[0067](6)(S21)
The instruction analyzing section 80 analyzes the command B, and m= 3 is calculated. Three data is written in output FIFO74.
[0068](7) (S22, S23)
num=24+3=27 is calculated in the data counting part 82. There is no read-out of data and it is r= 0. Therefore, it is num=27 (drawing 13).
[0069](8)(S24)
Since num>=28 is not satisfied, the Full check part 86 resets OFIFO_full_flag_m.
[0070](9)(S25)
Since OFIFO_full_flag_m is 0, control returns to S21 and processing of the next command (command C) is started.
[0071](10)(S21)
The instruction analyzing section 80 analyzes the command C, and m= 4 is calculated. Four data is written in output FIFO74.
[0072](11) (S22, S23)
num=num+m=27+4=31 is calculated in the data counting part 82 (drawing 14).
[0073](12)(S24)
Since num>=28 is realized, the Full check part 86 sets OFIFO_full_flag_m.
[0074](13)(S25)
Since OFIFO_full_flag_m is 1, the Full check part 86 repeats processing of S23-S25 until OFIFO_full_flag_m is set to one. If four or more data is read from output FIFO74, OFIFO_full_flag_m is reset and it is set to 0 (drawing 15), control will return to S21 and processing of the next command will be started further.
[0075]As mentioned above, with the device of this embodiment, it investigates whether output FIFO74 has free space, and if there is even sufficient free space, before all the data will be read, data can be written in output FIFO74. As a result, a possibility that the writing of output FIFO74 may stop can decrease and data transfer processing speed can be raised, using output FIFO74 efficiently.
[0076]According to Embodiment 2 described on Embodiment 3, the number of banks of output FIFO is 1, and improvement in data transfer processing speed is aimed at by investigating whether as so described above, there is any sufficient free space. However, the technique of investigating whether there is any sufficient free space for output FIFO is not a reason which can be applied only when the number of banks of output FIFO is 1, and also when the number of

banks is plurality, it can be applied. Embodiment 3 described below is such an example.
[0077]With reference to drawing 16, the processor 100 which has the output FIFO data transfer controller 112 concerning this Embodiment 3 is provided with the following.
Computing unit 40.
Output FIFO data transfer controller 112.
Output FIFO44 including two or more banks which outputs the data which receives from the output FIFO data transfer controller 112 to the output unit 46.

[0078]The output FIFO data transfer controller 112 is provided with the following.
The instruction analyzing section 120 for analyzing the command received from the computing unit 40.
The data counting part 122 for investigating whether output FIFO44 has free space.
The Full check part 126 for investigating whether output FIFO44 is in a Full state.

[0079]With reference to drawing 17, this output FIFO data transfer controller 112 operates as follows. If the write instruction of the data from the computing unit 40 to output FIFO44 is received, the instruction analyzing section 120 will ask for the data number (m) written in, and will write data in output FIFO44 (S21). The data counting part 122 adds the value m which the instruction analyzing section 120 calculated to the value num of the counter showing the data number stored in output FIFO44 which he is maintaining. That is, the following calculations are performed (S22).
[0080]num=num+m -- Further, when data is read from output FIFO44, only the read number (r) subtracts (4) data counting part 122 from the value num of a counter. That is, the following calculations are performed (S23).
[0081]num=num-r -- The value of the value num of the counter produced by calculating in this way of (5) data counting part 122 is the total size () of each bank of output FIFO44. [N*B and] However, B judges whether it is more than the value (NN=N*B-M) that subtracted from the number of banks the maximum number (M) of the data which can be transmitted with one command (S24). If it is num>N*B-M, flag OFIFO_full_flag_m will be set, and OFIFO_full_flag_m will be reset if it is except it (S34). If this is expressed in accordance with the notation of the C language, it will become as follows.
[0082]
OFIFO_full_flag_m=(num>=N*B-M) ? 1:0 -- (7) and the Full check part 126 repeat processing of S23-S25 until it will return to S23 (S25) and OFIFO_full_flag_m will be reset, if OFIFO_full_flag_m is 1. That is, the output FIFO data transfer controller 112 does not receive the next command until data is read from output FIFO44 and the conditional expression in a formula (6) stops concluding. If the conditional expression in a formula (6) is concluded, control will return to S21 and will process the next command.
[0083]The state of output FIFO44 at this time is explained concretely. As shown in drawing 18, output FIFO44 shall have the bank whose size is four of 32 respectively (B= 4). The number of the computing units 40 shall be four, therefore four data shall be simultaneously transmitted to output FIFO44 (M= 4). At this time, it is NN=N*B-M=32*4-4=124 of a formula (7). Therefore, OFIFO_full_flag_m is set at the time of num>=124.
[0084]For example, as shown in drawing 18, 121 data shall be written in output FIFO44, and one piece shall not be read, either (num=121). And it is assumed that it is that from which the following commands shall be published and output FIFO44 to two data is read between the

command A and the command B.
[0085]
A:1 command transmission (two-piece read-out)
According to concrete operation, the state of output FIFO74 changes as follows at the time of C:4 B:3 command transfer command ******.
[0086](1) (S21)
The command A is analyzed in the instruction analyzing section 120, and m= 1 is calculated. One data is written in output FIFO44.
[0087](2)(S22)
num=121+1=122 is calculated in the data counting part 122.
[0088](3)(S23)
It answers that two data was read and num=122-2=120 is calculated in the data counting part 122.
[0089](4)(S34)
Since num>=124 is not satisfied, the Full check part 126 resets OFIFO_full_flag_m.
[0090](5)(S25)
Since OFIFO_full_flag_m is 0, control returns to S21 and processing of the next command (command B) is started.
[0091](6)(S21)
The instruction analyzing section 120 analyzes the command B, and m= 3 is calculated. Three data is written in output FIFO44.
[0092](7) (S22, S23)
num=120+3=123 is calculated in the data counting part 122. There is no read-out of data and it is r= 0. Therefore, it is num=123.
[0093](8)(S24)
Since num>=124 is not satisfied, the Full check part 126 resets OFIFO_full_flag_m.
[0094](9)(S25)
Since OFIFO_full_flag_m is 0, control returns to S21 and processing of the next command (command C) is started.
[0095](10)(S21)
The instruction analyzing section 120 analyzes the command C, and m= 4 is calculated. Four data is written in output FIFO44.
[0096](11) (S22, S23)
num=num+m=123+4=127 is calculated in the data counting part 122.
[0097](12)(S24)
Since num>=124 is realized, the Full check part 126 sets OFIFO_full_flag_m.
[0098](13)(S25)
Since OFIFO_full_flag_m is 1, the Full check part 126 repeats processing of S23-S25 until OFIFO_full_flag_m is set to zero. If four or more data is read from output FIFO44, OFIFO_full_flag_m is reset and it is set to 0, control will return to S21 and processing of the next command will be started further.
[0099]Also when output FIFO44 has two or more banks, [in / as mentioned above / the device of this embodiment] If it judges whether there is any field sufficient for data transfer for output FIFO44 by counting the free space of output FIFO44 and there is sufficient field, data can be written in output FIFO44, without waiting to read all the data of output FIFO44. As a result, data transfer processing speed improvement can be carried out.

[0100]In the embodiment 4 embodiments 1-3, it is investigated whether the number of the data written in each bank of output FIFO was counted, and the size of the free space of output FIFO has satisfied predetermined conditions. There are also the following methods as a means for investigating this free space.
[0101]With reference to drawing 19, the processor 130 concerning this 4th embodiment is changed to the output FIFO data transfer controller 42 of a 1st embodiment shown in drawing 1, and contains the output FIFO data transfer controller 42 and the different output FIFO data transfer controller 132 in that it states below. The output FIFO data transfer controller 132 is changed to the data counting part 52 of the output FIFO data transfer controller 42, and differ in that it has the free space calculation part 144 which investigates the size of the free space of output FIFO44 in accordance with a method which is described below. In drawing 19, the same reference mark is given to the same parts as drawing 1. Those functions and names are also the same. Therefore, the detailed explanation about them is not repeated here.
[0102]With reference to drawing 20, operation of the output FIFO data transfer controller 132 is explained also including the function of the free space calculation part 144. First, the instruction analyzing section 50 computes the number of the data which analyzes a command and then should be written in output FIFO44 (m), and writes this data in output FIFO44 (S11). A free space calculation part subtracts the value of m from the value Counter of the counter in which the number of the data which can be written in each bank of output FIFO44 is shown (an initial value is Counter=N). That is, the value Counter of a counter is calculated by the following formulas (S42).
[0103]The Counter = Counter - m check flag generation part 54 compares the value Counter of a counter with the maximum number (M) of the data which can be written in output FIFO44 with one command. When the value Counter of a counter is less than M, the check flag generation part 54 sets the check flag fullcheck, and, in other cases, resets it (S13). If this is written in accordance with the notation of the C language, it will become as follows.
[0104]fullcheck = (Counter < M) ? The 1:0Full check part 56, A logical product with the check flag fullcheck which flag OFIFO_full_flag (set to 0 when other [it cannot do, and], 1 and) which shows whether the writing to the bank next to output FIFO44 is made, and the check flag generation part 54 generated is taken (S14). And as a result, it is judged whether OFIFO_full is 1 (S15).
[0105]Future processings are the same as that of the case of Embodiment 1. The same effect as Embodiment 1 can be acquired also by this Embodiment 4.
[0106]Same modification can be performed also with the embodiment 5 embodiment 2. That is, in Embodiment 2 shown in drawing 8, the data counting part 82 for counting the data number currently written in output FIFO was formed, and it is judged by the full check part 86 whether output FIFO74 is full based on the calculation result. The means for changing to this, computing directly the area size which can be further written in in output FIFO74, and maintaining it from the written-in data number and the read data number, is formed, It can have composition which compares the size in which the writing is possible with the maximum data number that can be written in at a time output FIFO74. The device of Embodiment 5 shown in drawing 21 is such a device. Since hardware composition is the same as that of what was shown in drawing 8 almost, here explains a control flow according to the flow chart of drawing 21. In the following explanation, what was shown in drawing 8 is used as a reference mark of each part. It shall change to the data counting part 82 of drawing 8, and shall have a free space calculation part (not shown).

[0107]With reference to drawing 21, this output FIFO data transfer controller operates as follows. If the write instruction of the data from the computing unit 40 to output FIFO74 is received, the instruction analyzing section 80 will ask for the data number (m) written in, and will write data in output FIFO74 (S21). A free space calculation part subtracts the value m which the instruction analyzing section 80 calculated from the value num of the counter showing the data number which can be written in in output FIFO74 which he is maintaining. That is, the following calculations are performed (S52).
[0108]Further, a num=num - m free space calculation part adds the read number (referred to as r.) to the value num of a counter, when data is read from output FIFO74. That is, the following calculations are performed (S53).
[0109]num=num+r -- The value of the value num of the counter produced by calculating (5) free-space calculation part in this way judges whether it is less than the maximum number (M) of the data which can be transmitted to output FIFO74 with one command (S24). If it is num<M, flag OFIFO_full_flag_m will be set, and OFIFO_full_flag_m will be reset if it is except it (S54). If this is expressed in accordance with the notation of the C language, it will become as follows.
[0110]
OFIFO_full_flag_m=(num<M) ? 1:0 -- (6) and the Full check part 86 repeat processing of S23-S25 until it will return to S23 (S25) and OFIFO_full_flag_m will be reset, if OFIFO_full_flag_m is 1.
[0111]Future processings are the same as that of the case of Embodiment 2, and can acquire the same effect.
[0112]The same idea is applicable also to Embodiment 3. However, the initial value of num is N*B in this case.
[0113]With all the points, the embodiment indicated this time is illustration and should be considered not to be restrictive. The range of this invention is shown by the above-mentioned not explanation but claim, and it is meant that a claim, an equivalent meaning, and all the change in within the limits are included.
[0114]
[Effect of the Invention]As mentioned above, according to this invention, since it is certainly judged whether there is any inconvenience although the following data is written in before reaching to the last of a bank, there is no possibility that overwrite of data may arise. It is not necessary to perform processing which prevents overwrite of data by a program, and improvement in data transfer processing speed can be aimed at.
[0115]When the availability more than the maximum data volume which can be transmitted at once is secured, or when the following bank is no longer full and processing of the next command is started, Since there is no possibility that it may be overwritten even if the data of a peak is transmitted to an output FIFO storage with the next command, it is not necessary to perform processing which prevents overwrite of data by a program, and improvement in data transfer processing speed can be aimed at.
[0116]Since according to other aspects of affairs of this invention the next command is processed after it is checked that there is certainly sufficient availability, there is no possibility that overwrite of data may arise. It is not necessary to perform processing which prevents overwrite of data by a program, and improvement in the speed of data transfer processing speed can be attained.
[0117]Even when the output FIFO storage includes two or more banks, and even when only the single bank is included, data transfer processing speed can be accelerated.

[0118]There is no possibility that overwrite of data may arise only at the time more than the maximum data volume which the availability of an output FIFO storage can transmit at once even if the data volume transmitted with the next command is the maximum data volume, if the next command is processed. Therefore, it is unnecessary to perform processing for avoiding overwrite of data in a program, and improvement in the speed of data transfer processing speed can be attained.

TECHNICAL FIELD

[Field of the Invention]In a processor with two or more computing units, this invention the transmission at the time of transmitting arbitrary numbers of data to an output unit via a pushup storage (only referred to as "FIFO" below.) with one command about the output FIFO data transfer controller for controlling especially, It is related with the output FIFO data transfer controller which can prevent overflow of FIFO.

PRIOR ART

[Description of the Prior Art]In the processor etc. which have two or more computing units, in order to absorb the difference of processing timing with an output unit, FIFO is used. The processor 140 concerning a Prior art with reference to drawing 22, The output of two or more computing units 40, the output FIFO data transfer controller 142 which undergoes the output of the computing unit 40, and the output FIFO data transfer controller 142 is held temporarily, and output FIFO44 for outputting to the output unit 46 is included.
[0003]The instruction analyzing section 150 for the output FIFO data transfer controller 142 to analyze the write instruction of output FIFO44, The data counting part 152 for counting the number of the data written in output FIFO44, The check flag generation part 154 for generating the flag for investigating whether output FIFO44 is Full and the Full check part 156 for judging whether output FIFO44 is in a Full state based on the Full flag which the check flag generation part 154 generated are included.
[0004]With reference to drawing 23, this output FIFO data transfer controller 142 operates as follows. First, the instruction analyzing section 150 analyzes the write-in command to output FIFO44 from the computing unit 40, and writes data in output FIFO44 (Step 1.). By the following explanation, a "step" is only indicated to be "S". . The data counting part 152 counts the data number written in output FIFO44 by what (S2) a count **********ed for whenever data is written in output FIFO44 by the instruction analyzing section 150. The check flag generation part 154 sets a check flag (fullcheck), when the value of the count of the data counting part 152 is more than the size of output FIFO44.
[0005]On the other hand, in output FIFO44, when the writing to the bank next to the bank under present writing is not made, by setting flag OFIFO_full_flag shows it. The Full check part 156 takes the logical product of OFIFO_full_flag and fullcheck, when the result is 1, judges with the writing of output FIFO44 being impossible, and sets the value of the flag fullcheck as 1 (S3). therefore, the Full check part 156 stops the writing of output FIFO44 by the next command until all the data of the following bank of output FIFO44 is read in this case, the following bank serves as empty, and OFIFO_full_flag is set to zero namely, -- (S5).

EFFECT OF THE INVENTION

[Effect of the Invention]As mentioned above, according to this invention, since it is certainly judged whether there is any inconvenience although the following data is written in before reaching to the last of a bank, there is no possibility that overwrite of data may arise. It is not necessary to perform processing which prevents overwrite of data by a program, and improvement in data transfer processing speed can be aimed at.
[0115]When the availability more than the maximum data volume which can be transmitted at once is secured, or when the following bank is no longer full and processing of the next command is started, Since there is no possibility that it may be overwritten even if the data of a peak is transmitted to an output FIFO storage with the next command, it is not necessary to perform processing which prevents overwrite of data by a program, and improvement in data transfer processing speed can be aimed at.
[0116]Since according to other aspects of affairs of this invention the next command is processed after it is checked that there is certainly sufficient availability, there is no possibility that overwrite of data may arise. It is not necessary to perform processing which prevents overwrite of data by a program, and improvement in the speed of data transfer processing speed can be attained.
[0117]Even when the output FIFO storage includes two or more banks, and even when only the single bank is included, data transfer processing speed can be accelerated.
[0118]There is no possibility that overwrite of data may arise only at the time more than the maximum data volume which the availability of an output FIFO storage can transmit at once even if the data volume transmitted with the next command is the maximum data volume, if the next command is processed. Therefore, it is unnecessary to perform processing for avoiding overwrite of data in a program, and improvement in the speed of data transfer processing speed can be attained.

TECHNICAL PROBLEM

[Problem(s) to be Solved by the Invention]In this conventional method, after data having been written in the end of the bank which has output FIFO44, for example, and having reached at the end of the bank is detected, it is judged whether data can be written in the following bank. That is, after data is written in in the end of a bank, it is judged whether the writing of the data to the following bank is possible.
[0007]Therefore, if the data transfer which must be stored ranging over two or more banks with one command arises in the conventional method, Also when the judgment that the writing to the following bank is impossible was made by the Full check part 156, there was a problem that the writing could not be stopped.
[0008]In order to solve such a problem, before performing processing which writes data in output FIFO44 by the program side which a processor executes, conventionally, the Full state of output FIFO44 always needed to be checked. For this check processing, when the conventional output FIFO data transfer controller is used, the fall of the data-processing transfer rate is caused.
[0009]For example, with reference to drawing 24, it is assumed that output FIFO44 is 4 bank configurations. The end of the bank 2 has an opening for one data so that it may be illustrated, but the bank 3 shall be write-in ending altogether. At this time, by the conventional method, when the data of the last of the bank 2 is written in, the FULL check is performed. Therefore, when it is the command that the command which writes in the data of the last of the bank 2, for

example transmits two or more data (for example, three data), in spite of not having read the data for two pieces of the bank 3 yet, it will be overwritten.
[0010]In order to keep such a problem from arising, it is necessary to confirm before transmission whether output FIFO44 is Full by the program side as mentioned above. A case may need to perform this check at every command issue each time.
[0011]One proposal for solving such a problem is indicated by JP,11-161467,A. In the art of an indication in this gazette, a memory is divided into two blocks on a certain boundary, and each is used as another FIFO. And while having prevented the circuitry of writing / read-out control circuit from devising the calculation (determination of next address) method of the next write-in position of each FIFO, and becoming complicated, It is judged whether free space is insufficient, respectively and whether the size and free space of data which are written in the next are compared for every FIFO, and it can write in.
[0012]However, in this art, since the number of FIFO is limited to two, mass data cannot be carried. Since two blocks are used as separate FIFO, there is a problem that the utilization ratio of a memory area is bad.
[0013]JP,63-167949,A is indicating the data transfer system with the sufficient efficiency which comprises two or more FIFO connected in series. In this art, it had the information which shows whether a FIFO buffer is empty for every FIFO, and grasps how much free space there is as the whole FIFO by it. And the size of free space and the size of data transmitting which were obtained in this way are compared, and only when the free space is larger, data is written in the FIFO buffer.
[0014]However, in this art, if it has the information which shows whether it is empty for every FIFO and at least one data is written in one FIFO, it will be supposed that it is not that FIFO empty. Therefore, the case where the field cannot be used also produces FIFO which still has free space actually. Therefore, the whole field of FIFO cannot be used efficiently too, but there is a problem that transfer efficiency also falls as a result.
[0015]so, the free space of FIFO according [the purpose of this invention] to a program -- it is providing the output FIFO data transfer controller which can raise data transfer processing speed by supposing that it is unnecessary and using a FIFO area efficiently.

MEANS

[Means for Solving the Problem]An output FIFO data transfer controller concerning this invention includes a command analysis circuit which writes the data concerned in an output FIFO storage while computing data volume transmitted by analyzing a data transfer instruction by a computing unit to an output FIFO storage which consists of two or more banks. An output FIFO storage has a function which outputs a full flag, when a bank next to a bank under writing is a full state. Further this output FIFO data transfer controller from data volume computed by command analysis circuit. Data count circuits which compute data volume written in a bank under output, judge whether an availability of a bank under output satisfies predetermined conditions, and output a decision flag, A full checking circuit which forbids processing of the next command is included until a full flag outputted from a decision flag or an output FIFO storage from data count circuits is reset.
[0017]An availability of a bank under output of an output FIFO storage is calculated after instruction processing, conditions with the availability are not satisfied, and when the following bank is a full state, processing of the next command is forbidden. Since it is judged whether there

is any inconvenience although the following data is written in before reaching to the last of a bank, there is no possibility that overwrite of data may arise. It is not necessary to judge whether it is full about each of two or more banks at this time. It is not necessary to perform processing which prevents overwrite of data by a program.

[0018]Preferably, data count circuits compute data volume written in a bank under output from data volume computed by command analysis circuit, judge whether conditions of being more than the maximum data volume which an availability of a bank under output can transmit at once are satisfied, and output a decision flag.

[0019]Since processing of the next command is started when an availability more than the maximum data volume which can be transmitted at once is secured, or when the following bank is no longer full, there is no possibility that it may be overwritten even if data of a peak is transmitted to an output FIFO storage with the next command.

[0020]According to other aspects of affairs of this invention, an output FIFO data transfer controller includes a command analysis circuit which writes the data concerned in an output FIFO storage while computing data volume transmitted by analyzing a data transfer instruction by a computing unit to an output FIFO storage. An output FIFO storage outputs a read signal which shows the number of data read when data was read from an output FIFO storage. Data volume with which this output FIFO data transfer controller was further computed by command analysis circuit, Data count circuits which compute an availability of an output FIFO storage, judge whether an availability of an output FIFO storage satisfies predetermined conditions, and output a decision flag from a read signal, A full checking circuit which forbids processing of the next command is included until a decision flag from data count circuits is reset.

[0021]An availability of an output FIFO storage is calculated after instruction processing, and when conditions with the availability are not satisfied, processing of the next command is forbidden. Since the next command is processed after it is checked that there is certainly sufficient availability, there is no possibility that overwrite of data may arise. It is not necessary to perform processing which prevents overwrite of data by a program, and improvement in the speed of processing can be attained.

[0022]An output FIFO storage may also include two or more banks, and may also include only a single bank. In any case, transmission processing of data is accelerable.

[0023]Data volume with which data count circuits were preferably computed by command analysis circuit, From a read signal, an availability of an output FIFO storage is computed, it judges whether conditions of being more than the maximum data volume which an availability of an output FIFO storage can transmit at once are satisfied, and a decision flag is outputted.

[0024]The next command is processed only at the time more than the maximum data volume which an availability of an output FIFO storage can transmit at once. Therefore, even if data volume transmitted with the next command is the maximum data volume, there is no possibility that overwrite of data may arise. Therefore, it is unnecessary to perform processing for avoiding overwrite of data in a program, and it can accelerate processing.

[0025]

[Embodiment of the Invention]With reference to embodiment 1 drawing 1, the processor 30 containing the output FIFO data transfer controller concerning the embodiment of the invention 1 is provided with the following.

Two or more computing units 40.

The output FIFO data transfer controller 42 which receives the data in which this computing unit 40 is outputted.

Output FIFO44 which consists of two or more banks for being controlled by the output FIFO data transfer controller 42, holding the data from the computing unit 40 temporarily, and transmitting to the output unit 46.

[0026]The output FIFO data transfer controller 42 is provided with the following.
The instruction analyzing section 50 which computes the data number (m) which analyzes the write-in command to output FIFO44, and is outputted to output FIFO44 with the command concerned.
The data counting part 52 for counting the data currently written in each bank of output FIFO44.
The check flag generation part 54 which generates the flag for investigating whether the bank of output FIFO44 is in a Full state.
The Full check part 56 for investigating whether the size of the free space of output FIFO44 has satisfied predetermined conditions, and forbidding the processing to the next command by the instruction analyzing section 50 according to the result.

[0027]With reference to drawing 2, this output FIFO data transfer controller 42 operates as follows. First, the instruction analyzing section 50 computes the number of the data which analyzes a command and then should be written in output FIFO44 (m) (S11), and writes this data in output FIFO44. The data counting part 52 adds the value of m to the value of the counter in which the number of the data currently written in each bank of output FIFO44 is shown. That is, the value Counter of a counter is calculated by the following formulas (S21).
[0028]Counter = Counter+m -- (1) check-flag generation part 54 compares the number (NN=N-M) which subtracted from the value Counter of a counter, and the size (N) of output FIFO44 the maximum number (M) of the data which can be written in with one command. When the value Counter of a counter is more than NN, the check flag generation part 54 sets the check flag fullcheck, and, in other cases, resets it (S13). If this is written in accordance with the notation of the C language, it will become as follows.
[0029]
fullcheck = (NN=(N-M))<=counter) ? 1:0 -- the (2) Full check part 56, A logical product with the check flag fullcheck which flag OFIFO_full_flag (set to 0 when other [it cannot do, and], 1 and) which shows whether the writing to the bank next to output FIFO44 is made, and the check flag generation part 54 generated is taken (S14). And as a result, it is judged whether OFIFO_full is 1 (S15). When result OFIFO_full is 1, processing returns to S14 (it is "YES" at S15). As a result, processing of S14 and S15 is repeated until the data of output FIFO44 is read and OFIFO_full_flag is reset, and processing of the command which writes in the next data of output FIFO44 is stopped. As for OFIFO_full_flag, N-1 in the bank of N FIFO shall be set, when data was written in, all the data is in the state which has not been read yet and data is written in the Nth bank for the first time.
[0030]If the result of a judgment of S15 serves as "NO", processing will return to S11, and processing to the next command is performed.
[0031]By having such composition, the number of the data written in output FIFO44 counts, before writing data in output FIFO44, and it is investigated in the end of a bank whether the size of the free space of output FIFO44 has always satisfied predetermined conditions. When data transfer which straddles the boundary of a bank occurs, it becomes unnecessary as a result, to be able to judge whether there is any possibility that a bank may serve as Full at high speed by hardware, and to judge whether a bank is Full by the program side. Software processing with

slow processing speed is avoided, and a data transfer rate can be raised.
[0032]A concrete example is given and operation of the output FIFO data transfer controller of this Embodiment 1 is explained.
[0033]As shown in drawing 3, output FIFO44 shall be [the size of each bank] 32 including four banks Bank 1-4. At this time, the value of NN=N-M shown in a formula (2) is NN=32-4=28. Therefore, when it comes to Counter>=28, fullcheck is set.
[0034]Data shall be finishing (unread appearance) in writing at the portion shown with the slash in drawing 3. That is, the banks 1, 3, and 4 shall be write-in ending altogether, only the bank 2 shall write them in by 25 pieces, they shall be ending, and shall be unread appearance. The number of computing units shall be four and a maximum of four data shall be simultaneously transmitted with one command. The bank under writing is the bank 2. Since the writing to the following bank (bank 3) is not made at this time, OFIFO_full_flag is set.
[0035]In this state, the following commands shall be published in order and the state of output FIFO44 at that time is explained.
[0036]
Data transfer (1) (S11: drawing 2) of C:4 data transfer instructions of B:3 data transfer instructions of A:1 command
The Full check part 56 analyzes the command A, and m= 1 is calculated. One data is written in output FIFO44 (drawing 4).
[0037](2)(S12)
Counter=Counter+1 is calculated in the data counting part 52. Therefore, Counter=25+1=26 is calculated.
[0038](3)(S13)
In the check flag generation part 54, since Countet>=28 is not realized, fullcheck is not set.
[0039](4) (S14, 15)
Since fullcheck is 0, processing returns to S11 as a result of the judgment in the check flag generation part 54.
[0040](5)(S11)
In the instruction analyzing section 50, the next command (command B) is analyzed and m= 3 is calculated. Furthermore, three data is written in to output FIFO44 (drawing 5).
[0041](6)(S12)
Counter=26+3=29 is calculated in the data counting part 52.
[0042](7)(S13)
In the check flag generation part 54, since Counter>=28 is realized, fullcheck is set.
[0043](8)(S14)
In the Full check part 56, since fullcheck is 1 also in 1 and OFIFO_full_flag, OFIFO_full is set to one.
[0044](9)(S15)
processing (S14 and S15) of the Full check part 56 is repeated [since OFIFO_full is 1,] until OFIFO_full_flag is reset, and the data of the bank 3 of output FIFO44 is read namely,. The data of the bank 3 is read, and if OFIFO_full_flag is reset, processing will return to S11 as a result of the judgment by the Full check part 56. The state of output FIFO44 at this time is shown in drawing 6.
[0045](10)(S11)
The instruction analyzing section 50 analyzes the next command C, calculates m= 4, and writes in four data to output FIFO44.

[0046](11)(S12)
Counter=29+4=33 is calculated in the data counting part 52.
[0047](12)(S13)
Since Counter>=28 is realized, the check flag generation part 54 sets fullcheck.
[0048](13)(S14)
OFIFO_full_flag is 0. Therefore, OFIFO_full which is a decision result of the Full check part 56 is also set to zero.
[0049](14)(S15)
Since OFIFO_full is 0, it returns to S11 (drawing 7). When data transfer is performed by next command, the data of the bank 3 is already read-out settled. Therefore, there is no possibility that overwrite of data may arise.
[0050]As mentioned above, according to the output FIFO data transfer controller 42 of this embodiment, when writing data in output FIFO44, it asks for that data number and the size of the free space of that bank was investigated in the end of a bank. And when a bank may be straddled and data may be written in, the next command is not received until it will be in the state where data transfer to output FIFO44 can be performed certainly. Therefore, even if the writing of data which straddles generates a bank, overflow can be avoided, and there is no possibility that it may be overwritten by the data which is not read. Therefore, it is not necessary to take the policy for preventing overwrite of such data by the program side. As a result, improvement in data transfer processing speed can be aimed at.
[0051]At Embodiment 1 described on Embodiment 2, it can transmit well with the composition which output FIFO44 consists of two or more banks, therefore was described above. However, OFIFO_full_flag will not be reset, if data is also partly written in even if it is a case where this bank has free space actually when only one piece has a bank and output FIFO tends to apply the same method as Embodiment 1. In the meantime, the writing of data stops. Therefore, there is a possibility that data transfer processing speed may fall.
[0052]The output FIFO data transfer controller explained in this Embodiment 2 is improved so that data transfer may be performed at high speed, even when the bank of output FIFO is one. Therefore, according to this Embodiment 2, it investigates how much free space not only the flag that shows [in which FIFO will only make it empty completely] whether it comes out but FIFO has, when there is free space, OFIFO_full_flag is reset, and when there is nothing, it sets. The output FIFO data transfer controller applied to this Embodiment 2 below is explained in detail.
[0053]With reference to drawing 8, the processor 70 containing the output FIFO data transfer controller 72 concerning this Embodiment 2 is provided with the following.
Computing unit 40.
Output FIFO data transfer controller 72.
Output FIFO74 for consisting of one bank and outputting the data from the output FIFO data transfer controller 72 to the output unit 46.

[0054]The output FIFO data transfer controller 72 is provided with the following.
The instruction analyzing section 80 for analyzing the write instruction of output FIFO74.
The data counting part 82 for investigating whether output FIFO74 has free space.
The Full check part 86 for investigating whether output FIFO74 is in a Full state.

[0055]With reference to drawing 9, the output FIFO data transfer controller 72 operates as follows. If the write instruction of the data from the computing unit 40 to output FIFO74 is

received, the instruction analyzing section 80 will ask for the data number (m) written in, and will write data in output FIFO74 (S21). The data counting part 82 adds the value m which the instruction analyzing section 80 calculated to the value num of the counter showing the data number stored in output FIFO74 which he is maintaining. That is, the following calculations are performed (S22).

[0056]num=num+m Further, --(4) data counting part 82 subtracts only the read number (referred to as r.) from the value num of a counter, when data is read from output FIFO74. That is, the following calculations are performed (S23).

[0057]num=num-r -- (5) data counting part 82 judges whether it is more than the value (N-M) in which the value of the value num of the counter produced by calculating in this way subtracted from the size (N) of output FIFO74 the maximum number (M) of the data which can be transmitted with one command (S24). If it is num>N-M, flag OFIFO_full_flag_m will be set, and OFIFO_full_flag_m will be reset if it is except it (S24). If this is expressed in accordance with the notation of the C language, it will become as follows.

[0058]

OFIFO_full_flag_m=(num>=N-M) ? 1:0 -- (6) and the Full check part 86 repeat processing of S23-S25 until it will return to S23 (S25) and OFIFO_full_flag_m will be reset, if OFIFO_full_flag_m is 1. That is, the output FIFO data transfer controller 72 does not receive the next command until data is read from output FIFO74 and the conditional expression in a formula (6) stops concluding. If the conditional expression in a formula (6) is concluded, control will return to S21 and will process the next command.

[0059]A concrete example explains operation of this output FIFO data transfer controller 72. As shown in drawing 10, size shall have one bank by 32 output FIFO74. As for the number of the computing units 40, four, therefore a maximum of four data shall be transmitted simultaneously. NN=N-M of a formula (6) is set to NN=32-4=8 at this time. Namely, the output FIFO data transfer controller 72 sets OFIFO_full_flag_m at the time of num>=28.

[0060]For example, as shown in drawing 10, 25 data is written in output FIFO74, and the case where one piece is not read, either is assumed (num=25). At this time, the following commands shall be published and two data shall be read between the command A and the command B.

[0061]

A:1 command transmission (two-piece read-out)

According to concrete operation, the state of output FIFO74 changes as follows at the time of C:4 B:3 command transfer command ******.

[0062](1) (S21)

The command A is analyzed in the instruction analyzing section 80, and m= 1 is calculated. One data is written in output FIFO74.

[0063](2)(S22)

num=25+1=26 is calculated in the data counting part 82 (drawing 11).

[0064](3)(S23)

It answers that two data was read and num=26-2=24 is calculated in the data counting part 82 (drawing 12).

[0065](4)(S24)

Since num>=28 is not satisfied, the Full check part 86 resets OFIFO_full_flag_m.

[0066](5)(S25)

Since OFIFO_full_flag_m is 0, control returns to S21 and processing of the next command (command B) is started.

[0067](6)(S21)
The instruction analyzing section 80 analyzes the command B, and m= 3 is calculated. Three data is written in output FIFO74.
[0068](7) (S22, S23)
num=24+3=27 is calculated in the data counting part 82. There is no read-out of data and it is r= 0. Therefore, it is num=27 (drawing 13).
[0069](8)(S24)
Since num>=28 is not satisfied, the Full check part 86 resets OFIFO_full_flag_m.
[0070](9)(S25)
Since OFIFO_full_flag_m is 0, control returns to S21 and processing of the next command (command C) is started.
[0071](10)(S21)
The instruction analyzing section 80 analyzes the command C, and m= 4 is calculated. Four data is written in output FIFO74.
[0072](11) (S22, S23)
num=num+m=27+4=31 is calculated in the data counting part 82 (drawing 14).
[0073](12)(S24)
Since num>=28 is realized, the Full check part 86 sets OFIFO_full_flag_m.
[0074](13)(S25)
Since OFIFO_full_flag_m is 1, the Full check part 86 repeats processing of S23-S25 until OFIFO_full_flag_m is set to one. If four or more data is read from output FIFO74, OFIFO_full_flag_m is reset and it is set to 0 (drawing 15), control will return to S21 and processing of the next command will be started further.
[0075]As mentioned above, with the device of this embodiment, it investigates whether output FIFO74 has free space, and if there is even sufficient free space, before all the data will be read, data can be written in output FIFO74. As a result, a possibility that the writing of output FIFO74 may stop can decrease and data transfer processing speed can be raised, using output FIFO74 efficiently.
[0076]According to Embodiment 2 described on Embodiment 3, the number of banks of output FIFO is 1, and improvement in data transfer processing speed is aimed at by investigating whether as so described above, there is any sufficient free space. However, the technique of investigating whether there is any sufficient free space for output FIFO is not a reason which can be applied only when the number of banks of output FIFO is 1, and also when the number of banks is plurality, it can be applied. Embodiment 3 described below is such an example.
[0077]With reference to drawing 16, the processor 100 which has the output FIFO data transfer controller 112 concerning this Embodiment 3 is provided with the following.
Computing unit 40.
Output FIFO data transfer controller 112.
Output FIFO44 including two or more banks which outputs the data which receives from the output FIFO data transfer controller 112 to the output unit 46.

[0078]The output FIFO data transfer controller 112 is provided with the following.
The instruction analyzing section 120 for analyzing the command received from the computing unit 40.
The data counting part 122 for investigating whether output FIFO44 has free space.
The Full check part 126 for investigating whether output FIFO44 is in a Full state.

[0079]With reference to drawing 17, this output FIFO data transfer controller 112 operates as follows. If the write instruction of the data from the computing unit 40 to output FIFO44 is received, the instruction analyzing section 120 will ask for the data number (m) written in, and will write data in output FIFO44 (S21). The data counting part 122 adds the value m which the instruction analyzing section 120 calculated to the value num of the counter showing the data number stored in output FIFO44 which he is maintaining. That is, the following calculations are performed (S22).
[0080]num=num+m -- Further, when data is read from output FIFO44, only the read number (r) subtracts (4) data counting part 122 from the value num of a counter. That is, the following calculations are performed (S23).
[0081]num=num-r -- The value of the value num of the counter produced by calculating in this way of (5) data counting part 122 is the total size () of each bank of output FIFO44. [N*B and] However, B judges whether it is more than the value (NN=N*B-M) that subtracted from the number of banks the maximum number (M) of the data which can be transmitted with one command (S24). If it is num>N*B-M, flag OFIFO_full_flag_m will be set, and OFIFO_full_flag_m will be reset if it is except it (S34). If this is expressed in accordance with the notation of the C language, it will become as follows.
[0082]
OFIFO_full_flag_m=(num>=N*B-M) ? 1:0 -- (7) and the Full check part 126 repeat processing of S23-S25 until it will return to S23 (S25) and OFIFO_full_flag_m will be reset, if OFIFO_full_flag_m is 1. That is, the output FIFO data transfer controller 112 does not receive the next command until data is read from output FIFO44 and the conditional expression in a formula (6) stops concluding. If the conditional expression in a formula (6) is concluded, control will return to S21 and will process the next command.
[0083]The state of output FIFO44 at this time is explained concretely. As shown in drawing 18, output FIFO44 shall have the bank whose size is four of 32 respectively (B= 4). The number of the computing units 40 shall be four, therefore four data shall be simultaneously transmitted to output FIFO44 (M= 4). At this time, it is NN=N*B-M=32*4-4=124 of a formula (7). Therefore, OFIFO_full_flag_m is set at the time of num>=124.
[0084]For example, as shown in drawing 18, 121 data shall be written in output FIFO44, and one piece shall not be read, either (num=121). And it is assumed that it is that from which the following commands shall be published and output FIFO44 to two data is read between the command A and the command B.
[0085]
A:1 command transmission (two-piece read-out)
According to concrete operation, the state of output FIFO74 changes as follows at the time of C:4 B:3 command transfer command ******.
[0086](1) (S21)
The command A is analyzed in the instruction analyzing section 120, and m= 1 is calculated. One data is written in output FIFO44.
[0087](2)(S22)
num=121+1=122 is calculated in the data counting part 122.
[0088](3)(S23)
It answers that two data was read and num=122-2=120 is calculated in the data counting part 122.

[0089](4)(S34)
Since num>=124 is not satisfied, the Full check part 126 resets OFIFO_full_flag_m.
[0090](5)(S25)
Since OFIFO_full_flag_m is 0, control returns to S21 and processing of the next command (command B) is started.
[0091](6)(S21)
The instruction analyzing section 120 analyzes the command B, and m= 3 is calculated. Three data is written in output FIFO44.
[0092](7) (S22, S23)
num=120+3=123 is calculated in the data counting part 122. There is no read-out of data and it is r= 0. Therefore, it is num=123.
[0093](8)(S24)
Since num>=124 is not satisfied, the Full check part 126 resets OFIFO_full_flag_m.
[0094](9)(S25)
Since OFIFO_full_flag_m is 0, control returns to S21 and processing of the next command (command C) is started.
[0095](10)(S21)
The instruction analyzing section 120 analyzes the command C, and m= 4 is calculated. Four data is written in output FIFO44.
[0096](11) (S22, S23)
num=num+m=123+4=127 is calculated in the data counting part 122.
[0097](12)(S24)
Since num>=124 is realized, the Full check part 126 sets OFIFO_full_flag_m.
[0098](13)(S25)
Since OFIFO_full_flag_m is 1, the Full check part 126 repeats processing of S23-S25 until OFIFO_full_flag_m is set to zero. If four or more data is read from output FIFO44, OFIFO_full_flag_m is reset and it is set to 0, control will return to S21 and processing of the next command will be started further.
[0099]Also when output FIFO44 has two or more banks, [in / as mentioned above / the device of this embodiment] If it judges whether there is any field sufficient for data transfer for output FIFO44 by counting the free space of output FIFO44 and there is sufficient field, data can be written in output FIFO44, without waiting to read all the data of output FIFO44. As a result, data transfer processing speed improvement can be carried out.
[0100]In the embodiment 4 embodiments 1-3, it is investigated whether the number of the data written in each bank of output FIFO was counted, and the size of the free space of output FIFO has satisfied predetermined conditions. There are also the following methods as a means for investigating this free space.
[0101]With reference to drawing 19, the processor 130 concerning this 4th embodiment is changed to the output FIFO data transfer controller 42 of a 1st embodiment shown in drawing 1, and contains the output FIFO data transfer controller 42 and the different output FIFO data transfer controller 132 in that it states below. The output FIFO data transfer controller 132 is changed to the data counting part 52 of the output FIFO data transfer controller 42, and differ in that it has the free space calculation part 144 which investigates the size of the free space of output FIFO44 in accordance with a method which is described below. In drawing 19, the same reference mark is given to the same parts as drawing 1. Those functions and names are also the same. Therefore, the detailed explanation about them is not repeated here.

[0102]With reference to drawing 20, operation of the output FIFO data transfer controller 132 is explained also including the function of the free space calculation part 144. First, the instruction analyzing section 50 computes the number of the data which analyzes a command and then should be written in output FIFO44 (m), and writes this data in output FIFO44 (S11). A free space calculation part subtracts the value of m from the value Counter of the counter in which the number of the data which can be written in each bank of output FIFO44 is shown (an initial value is Counter=N). That is, the value Counter of a counter is calculated by the following formulas (S42).

[0103]The Counter = Counter - m check flag generation part 54 compares the value Counter of a counter with the maximum number (M) of the data which can be written in output FIFO44 with one command. When the value Counter of a counter is less than M, the check flag generation part 54 sets the check flag fullcheck, and, in other cases, resets it (S13). If this is written in accordance with the notation of the C language, it will become as follows.

[0104]fullcheck = (Counter < M) ? The 1:0Full check part 56, A logical product with the check flag fullcheck which flag OFIFO_full_flag (set to 0 when other [it cannot do, and], 1 and) which shows whether the writing to the bank next to output FIFO44 is made, and the check flag generation part 54 generated is taken (S14). And as a result, it is judged whether OFIFO_full is 1 (S15).

[0105]Future processings are the same as that of the case of Embodiment 1. The same effect as Embodiment 1 can be acquired also by this Embodiment 4.

[0106]Same modification can be performed also with the embodiment 5 embodiment 2. That is, in Embodiment 2 shown in drawing 8, the data counting part 82 for counting the data number currently written in output FIFO was formed, and it is judged by the full check part 86 whether output FIFO74 is full based on the calculation result. The means for changing to this, computing directly the area size which can be further written in in output FIFO74, and maintaining it from the written-in data number and the read data number, is formed, It can have composition which compares the size in which the writing is possible with the maximum data number that can be written in at a time output FIFO74. The device of Embodiment 5 shown in drawing 21 is such a device. Since hardware composition is the same as that of what was shown in drawing 8 almost, here explains a control flow according to the flow chart of drawing 21. In the following explanation, what was shown in drawing 8 is used as a reference mark of each part. It shall change to the data counting part 82 of drawing 8, and shall have a free space calculation part (not shown).

[0107]With reference to drawing 21, this output FIFO data transfer controller operates as follows. If the write instruction of the data from the computing unit 40 to output FIFO74 is received, the instruction analyzing section 80 will ask for the data number (m) written in, and will write data in output FIFO74 (S21). A free space calculation part subtracts the value m which the instruction analyzing section 80 calculated from the value num of the counter showing the data number which can be written in in output FIFO74 which he is maintaining. That is, the following calculations are performed (S52).

[0108]Further, a num=num - m free space calculation part adds the read number (referred to as r.) to the value num of a counter, when data is read from output FIFO74. That is, the following calculations are performed (S53).

[0109]num=num+r -- The value of the value num of the counter produced by calculating (5) free-space calculation part in this way judges whether it is less than the maximum number (M) of the data which can be transmitted to output FIFO74 with one command (S24). If it is num<M,

flag OFIFO_full_flag_m will be set, and OFIFO_full_flag_m will be reset if it is except it (S54). If this is expressed in accordance with the notation of the C language, it will become as follows.
[0110]
OFIFO_full_flag_m=(num<M) ? 1:0 -- (6) and the Full check part 86 repeat processing of S23-S25 until it will return to S23 (S25) and OFIFO_full_flag_m will be reset, if OFIFO_full_flag_m is 1.
[0111]Future processings are the same as that of the case of Embodiment 2, and can acquire the same effect.
[0112]The same idea is applicable also to Embodiment 3. However, the initial value of num is N*B in this case.
[0113]With all the points, the embodiment indicated this time is illustration and should be considered not to be restrictive. The range of this invention is shown by the above-mentioned not explanation but claim, and it is meant that a claim, an equivalent meaning, and all the change in within the limits are included.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]
[Drawing 1]It is a block diagram of the processor containing the output FIFO data transfer controller concerning the embodiment of the invention 1.
[Drawing 2]It is a flow chart of the control program of the output FIFO data transfer controller concerning the embodiment of the invention 1.
[Drawing 3]It is a mimetic diagram of FIFO for operation of the output FIFO data transfer controller concerning the embodiment of the invention 1 to be shown.
[Drawing 4]It is a mimetic diagram of the bank 2 of FIFO for operation of the output FIFO data transfer controller concerning the embodiment of the invention 1 to be shown.
[Drawing 5]It is a mimetic diagram of the bank 2 of FIFO for operation of the output FIFO data transfer controller concerning the embodiment of the invention 1 to be shown.
[Drawing 6]It is a mimetic diagram of the bank 2 of FIFO for operation of the output FIFO data transfer controller concerning the embodiment of the invention 1 to be shown.
[Drawing 7]It is a mimetic diagram of the bank 2 of FIFO for operation of the output FIFO data transfer controller concerning the embodiment of the invention 1 to be shown.
[Drawing 8]It is a block diagram of the processor containing the output FIFO data transfer controller concerning the embodiment of the invention 2.
[Drawing 9]It is a flow chart of the control program of the output FIFO data transfer controller concerning the embodiment of the invention 2.
[Drawing 10]It is a mimetic diagram of the bank 2 of FIFO for operation of the output FIFO data transfer controller concerning the embodiment of the invention 2 to be shown.
[Drawing 11]It is a mimetic diagram of the bank 2 of FIFO for operation of the output FIFO data transfer controller concerning the embodiment of the invention 2 to be shown.
[Drawing 12]It is a mimetic diagram of the bank 2 of FIFO for operation of the output FIFO data transfer controller concerning the embodiment of the invention 2 to be shown.
[Drawing 13]It is a mimetic diagram of the bank 2 of FIFO for operation of the output FIFO data transfer controller concerning the embodiment of the invention 2 to be shown.
[Drawing 14]It is a mimetic diagram of the bank 2 of FIFO for operation of the output FIFO data transfer controller concerning the embodiment of the invention 2 to be shown.

[Drawing 15]It is a mimetic diagram of the bank 2 of FIFO for operation of the output FIFO data transfer controller concerning the embodiment of the invention 2 to be shown.
[Drawing 16]It is a block diagram of the processor containing the output FIFO data transfer controller concerning the embodiment of the invention 3.
[Drawing 17]It is a flow chart of the control program of the output FIFO data transfer controller concerning the embodiment of the invention 3.
[Drawing 18]It is a mimetic diagram of FIFO for operation of the output FIFO data transfer controller concerning the embodiment of the invention 3 to be shown.
[Drawing 19]It is a block diagram of the processor containing the output FIFO data transfer controller concerning the embodiment of the invention 4.
[Drawing 20]It is a flow chart of the control program of the output FIFO data transfer controller concerning the embodiment of the invention 4.
[Drawing 21]It is a flow chart of the control program of the output FIFO data transfer controller concerning the embodiment of the invention 5.
[Drawing 22]It is a block diagram of the processor containing the output FIFO data transfer controller concerning a Prior art.
[Drawing 23]It is a flow chart of the control program of the output FIFO data transfer controller concerning a Prior art.
[Drawing 24]It is a mimetic diagram of FIFO for operation of the output FIFO data transfer controller concerning a Prior art to be shown.
[Description of Notations]
30 and 70,100,130,140 A processor and 40 Computing unit, 42, 72,112,132,142 output FIFO data transfer controllers, 50 and 80,120,150 An instruction analyzing section, and 52 and 82,122,152 A data counting part and 54,154 A check flag generation part, 56, an 86,126,156 Full check part, 134 free-space calculation part.

[Translation done.]

(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号
特開2001－290700
（P2001－290700A）
(43)公開日　平成13年10月19日(2001.10.19)

(51)Int.Cl.[7]		識別記号	FI		テーマコード(参考)
Ｇ０６Ｆ	12/02	５３０	Ｇ０６Ｆ 12/02	５３０Ｄ	５Ｂ０６０
		５８０		５８０Ａ	５Ｂ０７７
	5/06	３１１	5/06	３１１	
	12/00	５６０	12/00	５６０Ｂ	
		５９４		５９４	

審査請求　未請求　請求項の数６　ＯＬ　（全 14 頁）　最終頁に続く

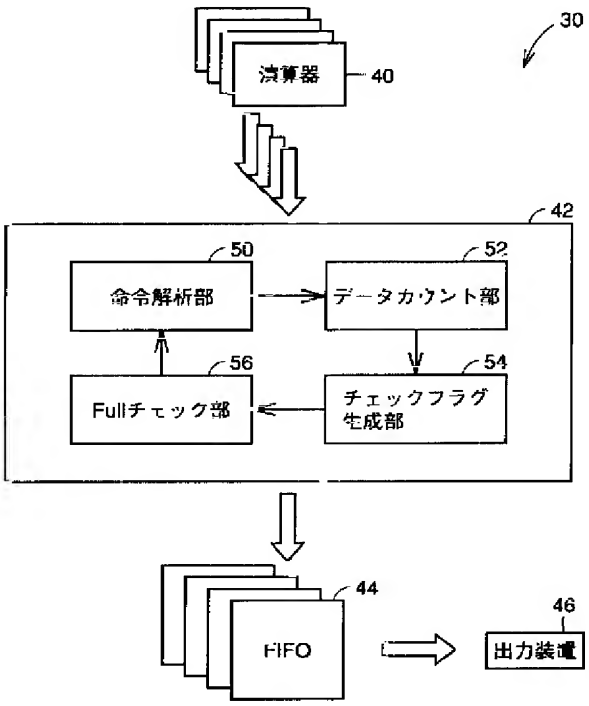
(21)出願番号　特願2000－106780(P2000－106780)
(22)出願日　平成12年４月７日(2000.4.7)

(71)出願人　000006013
三菱電機株式会社
東京都千代田区丸の内二丁目２番３号
(72)発明者　小原　淳子
東京都千代田区丸の内二丁目２番３号　三菱電機株式会社内
(72)発明者　河合　浩行
東京都千代田区丸の内二丁目２番３号　三菱電機株式会社内
(74)代理人　100064746
弁理士　深見　久郎　（外４名）

最終頁に続く

(54)【発明の名称】　出力ＦＩＦＯデータ転送制御装置

(57)【要約】
【課題】　プログラムによるＦＩＦＯの空き領域の確認を不要とし、効率良くＦＩＦＯ領域を利用することによってデータ転送処理速度を向上させることが可能な出力ＦＩＦＯデータ転送制御装置を提供すること。
【解決手段】　出力ＦＩＦＯデータ転送制御装置４２は、複数個のバンクを含む出力ＦＩＦＯ記憶装置４４へのデータ転送命令を解析し、転送されるデータ量を算出する命令解析部５０と、命令解析部５０により算出されたデータ量から出力中のバンクに書込まれたデータ量を算出し、出力中のバンクの空き容量が所定の条件を充足するか否かを示す判定フラグを出力するデータカウント部５２、５４と、データカウント部５２、５４からの判定フラグまたは出力ＦＩＦＯ記憶装置４４から出力されるフルフラグがリセットされるまで次の命令の処理を禁止するフルチェック部５６とを含む。



【特許請求の範囲】
【請求項１】　複数個のバンクからなり、書込中のバンクの次のバンクがフル状態のときにはフルフラグを出力する機能を有する出力ＦＩＦＯ記憶装置への演算器によるデータ転送命令を解析して、転送されるデータ量を算出するとともに当該データを前記出力ＦＩＦＯ記憶装置に書込む命令解析回路と前記命令解析回路により算出されたデータ量から、出力中のバンクに書込まれたデータ量を算出し、前記出力中のバンクの空き容量が所定の条件を充足するか否かを判定して判定フラグを出力するデータカウント回路と、
前記データカウント回路からの判定フラグまたは前記出力ＦＩＦＯ記憶装置から出力される前記フルフラグがリセットされるまで次の命令の処理を禁止するフルチェック回路とを含む、出力ＦＩＦＯデータ転送制御装置。
【請求項２】　前記データカウント回路は、前記命令解析回路により算出されたデータ量から、出力中のバンクに書込まれたデータ量を算出し、前記出力中のバンクの空き容量が一回に転送可能な最大データ量以上であるという条件が充足されるか否かを判定して判定フラグを出力する、請求項１に記載の出力ＦＩＦＯデータ転送制御装置。
【請求項３】　データが読み出されると読み出されたデータの個数を示す読出信号を出力する出力ＦＩＦＯ記憶装置への演算器によるデータ転送命令を解析して、転送されるデータ量を算出するとともに当該データを前記出力ＦＩＦＯ記憶装置に書込む命令解析回路と、
前記命令解析回路により算出されたデータ量と、前記読出信号とから、前記出力ＦＩＦＯ記憶装置の空き容量を算出し、前記出力ＦＩＦＯ記憶装置の空き容量が所定の条件を充足するか否かを判定して判定フラグを出力するデータカウント回路と、
前記データカウント回路からの判定フラグがリセットされるまで次の命令の処理を禁止するフルチェック回路とを含む、出力ＦＩＦＯデータ転送制御装置。
【請求項４】　前記出力ＦＩＦＯ記憶装置は複数個のバンクを含む、請求項３に記載の出力ＦＩＦＯデータ転送制御装置。
【請求項５】　前記出力ＦＩＦＯ記憶装置は単一のバンクを含む、請求項３に記載の出力ＦＩＦＯデータ転送制御装置。
【請求項６】　前記データカウント回路は、前記命令解析回路により算出されたデータ量と、前記読出信号とから、前記出力ＦＩＦＯ記憶装置の空き容量を算出し、前記出力ＦＩＦＯ記憶装置の空き容量が１回に転送可能な最大データ量以上であるという条件が充足されるか否かを判定して判定フラグを出力する、請求項３～請求項５のいずれかに記載の出力ＦＩＦＯデータ転送制御装置。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】この発明は、複数個の演算器を持つプロセッサにおいて、１つの命令により任意の数のデータを先入先出記憶装置（以下単に「ＦＩＦＯ」と呼ぶ。）を介して出力装置に転送する際の転送を制御するための出力ＦＩＦＯデータ転送制御装置に関し、特に、ＦＩＦＯのオーバーフローを防止することが可能な出力ＦＩＦＯデータ転送制御装置に関する。
【０００２】
【従来の技術】複数個の演算器を有するプロセッサなどにおいて、出力装置との処理タイミングの相違を吸収したりするために、ＦＩＦＯが使用される。図２２を参照して、従来の技術にかかるプロセッサ１４０は、複数個の演算器４０と、演算器４０の出力を受ける出力ＦＩＦＯデータ転送制御装置１４２と、出力ＦＩＦＯデータ転送制御装置１４２の出力を一時保持し、出力装置４６に出力するための出力ＦＩＦＯ４４とを含む。
【０００３】出力ＦＩＦＯデータ転送制御装置１４２は、出力ＦＩＦＯ４４への書込み命令を解析するための命令解析部１５０と、出力ＦＩＦＯ４４に書込まれたデータの数をカウントするためのデータカウント部１５２と、出力ＦＩＦＯ４４がＦｕｌｌか否かを調べるためのフラグを生成するためのチェックフラグ生成部１５４と、チェックフラグ生成部１５４が生成したＦｕｌｌフラグに基づいて出力ＦＩＦＯ４４がＦｕｌｌ状態か否かを判定するためのＦｕｌｌチェック部１５６とを含む。
【０００４】図２３を参照して、この出力ＦＩＦＯデータ転送制御装置１４２は以下のように動作する。まず命令解析部１５０が、演算器４０からの出力ＦＩＦＯ４４への書込命令を解析し、出力ＦＩＦＯ４４にデータを書込む（ステップ１。なお以下の説明では「ステップ」を単に「Ｓ」と記載する。）。データカウント部１５２は、命令解析部１５０によって出力ＦＩＦＯ４４にデータが書込まれるたびにカウントをインクリメントする（Ｓ２）ことにより出力ＦＩＦＯ４４に書込まれたデータ数をカウントする。チェックフラグ生成部１５４は、データカウント部１５２のカウントの値が出力ＦＩＦＯ４４のサイズ以上の場合にはチェックフラグ（fullcheck）をセットする。
【０００５】一方出力ＦＩＦＯ４４においては、現在書込中のバンクの次のバンクへの書込ができないときには、フラグOFIFO_full_flagをセットすることによりそれを示す。Ｆｕｌｌチェック部１５６は、OFIFO_full_flagとfullcheckとの論理積をとり、その結果が１の場合には出力ＦＩＦＯ４４への書込が不可であると判定してフラグfullcheckの値を１に設定する（Ｓ３）。したがってこの場合、Ｆｕｌｌチェック部１５６は、出力ＦＩＦＯ４４の次のバンクのデータが全て読み出され次のバンクが空となるまで（すなわちOFIFO_full_flagが０になるまで）、次の命令による出力ＦＩＦＯ４４への書込を止める（Ｓ５）。

【０００６】
【発明が解決しようとする課題】この従来の方法では、たとえば出力ＦＩＦＯ４４のあるバンクの最後にデータが書込まれてバンクの最後に達したことが検出された後に、次のバンクにデータを書込むことができるか否かを判定している。すなわち、バンクの終わりにデータが書込まれた後で、次のバンクへのデータの書込が可能か否かが判定される。
【０００７】そのため、従来の方法では、１命令で複数のバンクにまたがって格納しなければならないデータ転送が生じると、Ｆｕｌｌチェック部１５６によって次のバンクへの書込が不可能であるという判定がなされている場合にも、その書込を止めることができないという問題があった。
【０００８】こうした問題を解決するために従来は、プロセッサが実行するプログラム側で、出力ＦＩＦＯ４４にデータを書込む処理を行なう前に、必ず出力ＦＩＦＯ４４のＦｕｌｌ状態をチェックする必要があった。このチェック処理のため、従来の出力ＦＩＦＯデータ転送制御装置を用いた場合にはデータ処理転送速度の低下が引き起こされている。
【０００９】たとえば、図２４を参照して、出力ＦＩＦＯ４４が4バンク構成であると想定する。図示されるように、バンク２の最後にデータ１個分の空きがあるが、バンク３は全て書込み済みであるものとする。このとき、従来の方法では、バンク２の最後のデータが書込まれた時点でＦＵＬＬチェックを行なっている。そのためたとえばバンク２の最後のデータを書込む命令が２個以上のデータ（たとえば３個のデータ）を転送するような命令であった場合には、バンク３の２個分のデータが、まだ読み出されていないにもかかわらず上書きされてしまうことになる。
【００１０】こうした問題が生じないようにするためには、前述したとおりプログラム側で転送の前に出力ＦＩＦＯ４４がＦｕｌｌか否かをチェックする必要がある。場合によっては、命令発行のたびに毎回このチェックを行なう必要が生じることもある。
【００１１】こうした問題を解決するための一つの提案が特開平１１－１６１４６７号公報に開示されている。この公報に開示の技術では、メモリをある境界で二つのブロックに分け、それぞれを別のＦＩＦＯとして使用している。そしてそれぞれのＦＩＦＯの次の書込み位置の算出（次アドレスの決定）方法を工夫して書込み／読出制御回路の回路構成が複雑になるのを防止しているとともに、各ＦＩＦＯごとに、次に書込まれるデータの大きさと空き領域とを比較して、それぞれ空き領域が不足しているか否か、書込が可能か否かを判定している。
【００１２】しかしこの技術では、ＦＩＦＯの数が２つに限定されているために、大容量のデータを扱うことができない。また、二つのブロックが別々のＦＩＦＯとして使用されているため、メモリ領域の使用効率が悪いという問題がある。
【００１３】また、特開昭６３－１６７９４９号公報は、直列に接続された複数個のＦＩＦＯで構成される効率のよいデータ転送システムを開示している。この技術では、ＦＩＦＯ毎に、ＦＩＦＯバッファがエンプティか否かを示す情報を持ち、それによってＦＩＦＯ全体としてどの程度の空き領域があるかを把握している。そして、こうして得られた空き領域の大きさと転送データの大きさとを比較して空き領域の方が大きい場合にのみデータをＦＩＦＯバッファに書込んでいる。
【００１４】しかしこの技術では、ＦＩＦＯ毎に、エンプティか否かを示す情報を持ち、かつ一つのＦＩＦＯに一つでもデータが書込まれるとそのＦＩＦＯはエンプティでないとされる。そのため、実際にはまだ空き領域をもつＦＩＦＯでもその領域を使用することができない場合が生ずる。そのためにやはりＦＩＦＯの領域全体を効率良く使用できずその結果転送効率も低下するという問題がある。
【００１５】それゆえに本発明の目的は、プログラムによるＦＩＦＯの空き領域不要とし、効率良くＦＩＦＯ領域を利用することによってデータ転送処理速度を向上させることが可能な出力ＦＩＦＯデータ転送制御装置を提供することである。
【００１６】
【課題を解決するための手段】この発明にかかる出力ＦＩＦＯデータ転送制御装置は、複数個のバンクからなる出力ＦＩＦＯ記憶装置への演算器によるデータ転送命令を解析して、転送されるデータ量を算出するとともに当該データを出力ＦＩＦＯ記憶装置に書込む命令解析回路を含む。出力ＦＩＦＯ記憶装置は、書込中のバンクの次のバンクがフル状態のときにはフルフラグを出力する機能を有する。この出力ＦＩＦＯデータ転送制御装置はさらに、命令解析回路により算出されたデータ量から、出力中のバンクに書込まれたデータ量を算出し、出力中のバンクの空き容量が所定の条件を充足するか否かを判定して判定フラグを出力するデータカウント回路と、データカウント回路からの判定フラグまたは出力ＦＩＦＯ記憶装置から出力されるフルフラグがリセットされるまで次の命令の処理を禁止するフルチェック回路とを含む。
【００１７】命令処理後、出力ＦＩＦＯ記憶装置の出力中のバンクの空き容量を計算し、その空き容量がある条件を充足しておらず、かつ次のバンクがフル状態である場合には、次の命令の処理が禁止される。バンクの最後まで到達する前に、次のデータを書込んでも不都合がないか否かが判定されるため、データの上書きが生ずるおそれはない。このとき、複数個のバンクの各々についてフルか否かを判定する必要はない。またプログラムでデータの上書きを防止する処理を行なう必要もない。
【００１８】好ましくは、データカウント回路は、命令

解析回路により算出されたデータ量から、出力中のバンクに書込まれたデータ量を算出し、出力中のバンクの空き容量が一回に転送可能な最大データ量以上であるという条件が充足されるか否かを判定して判定フラグを出力する。

【００１９】一回に転送可能な最大データ量以上の空き容量が確保されたとき、または次のバンクがフルでなくなったときに次の命令の処理が開始されるので、次の命令で最大量のデータが出力ＦＩＦＯ記憶装置に転送されても上書きされたりするおそれはない。

【００２０】この発明の他の局面によれば、出力ＦＩＦＯデータ転送制御装置は、出力ＦＩＦＯ記憶装置への演算器によるデータ転送命令を解析して、転送されるデータ量を算出するとともに当該データを出力ＦＩＦＯ記憶装置に書込む命令解析回路を含む。出力ＦＩＦＯ記憶装置は、出力ＦＩＦＯ記憶装置からデータが読み出されると読み出されたデータの個数を示す読出信号を出力する。この出力ＦＩＦＯデータ転送制御装置はさらに、命令解析回路により算出されたデータ量と、読出信号とから、出力ＦＩＦＯ記憶装置の空き容量を算出し、出力ＦＩＦＯ記憶装置の空き容量が所定の条件を充足するか否かを判定して判定フラグを出力するデータカウント回路と、データカウント回路からの判定フラグがリセットされるまで次の命令の処理を禁止するフルチェック回路とを含む。

【００２１】命令処理後、出力ＦＩＦＯ記憶装置の空き容量を計算し、その空き容量がある条件を充足していない場合には、次の命令の処理が禁止される。必ず十分な空き容量があることが確認されてから次の命令が処理されるため、データの上書きが生ずるおそれはない。またプログラムでデータの上書きを防止する処理を行なう必要もなく、処理の高速化を図ることができる。

【００２２】出力ＦＩＦＯ記憶装置は複数個のバンクを含んでもよいし、単一のバンクのみ含んでもよい。いずれの場合にも、データの転送処理を高速化することができる。

【００２３】好ましくは、データカウント回路は、命令解析回路により算出されたデータ量と、読出信号とから、出力ＦＩＦＯ記憶装置の空き容量を算出し、出力ＦＩＦＯ記憶装置の空き容量が１回に転送可能な最大データ量以上であるという条件が充足されるか否かを判定して判定フラグを出力する。

【００２４】出力ＦＩＦＯ記憶装置の空き容量が１回に転送可能な最大データ量以上のときのみ次の命令が処理される。したがって、次の命令で転送されるデータ量が最大データ量であったとしてもデータの上書きが生ずるおそれはない。そのためプログラムにおいてデータの上書きを回避するための処理を行なうことが不要で、処理を高速化することができる。

【００２５】

【発明の実施の形態】実施の形態１

図１を参照して、本発明の実施の形態１にかかる出力ＦＩＦＯデータ転送制御装置を含むプロセッサ３０は、複数個の演算器４０と、この演算器４０の出力されるデータを受ける出力ＦＩＦＯデータ転送制御装置４２と、出力ＦＩＦＯデータ転送制御装置４２によって制御され、演算器４０からのデータを一時保持して出力装置４６に転送するための、複数個のバンクからなる出力ＦＩＦＯ４４とを含む。

【００２６】出力ＦＩＦＯデータ転送制御装置４２は、出力ＦＩＦＯ４４への書込命令を解析し、当該命令によって出力ＦＩＦＯ４４に出力されるデータ数（ｍ）を算出する命令解析部５０と、出力ＦＩＦＯ４４の各バンクに書込まれているデータをカウントするためのデータカウント部５２と、出力ＦＩＦＯ４４のバンクがＦｕｌｌ状態か否かを調べるためのフラグを生成するチェックフラグ生成部５４と、出力ＦＩＦＯ４４の空き領域の大きさが所定の条件を充足しているか否かを調べ、その結果に応じて命令解析部５０による次の命令に対する処理を禁止するためのＦｕｌｌチェック部５６とを含む。

【００２７】図２を参照して、この出力ＦＩＦＯデータ転送制御装置４２は以下のように動作する。まず命令解析部５０は、命令を解析して次に出力ＦＩＦＯ４４に書込まれるべきデータの数（ｍ）を算出し（Ｓ１１）、このデータを出力ＦＩＦＯ４４に書込む。データカウント部５２は、出力ＦＩＦＯ４４の各バンクに書込まれているデータの数を示すカウンタの値にｍの値を加算する。すなわち、以下の式によってカウンタの値Counterを計算する（Ｓ２１）。

【００２８】Counter = Counter + m　…(1)

チェックフラグ生成部５４は、カウンタの値Counterと、出力ＦＩＦＯ４４のサイズ（Ｎ）から１命令で書込めるデータの最大数（Ｍ）を減算した数（ＮＮ＝Ｎ－Ｍ）とを比較する。チェックフラグ生成部５４は、カウンタの値CounterがＮＮ以上の場合には、チェックフラグfullcheckをセットし、他の場合にはリセットする（Ｓ１３）。これをＣ言語の記法にしたがって書けば以下のとおりとなる。

【００２９】

fullcheck = (NN=(N-M))<=counter) ? 1:0　…(2)

Ｆｕｌｌチェック部５６は、出力ＦＩＦＯ４４の次のバンクへの書込みができるか否かを示すフラグOFIFO_full_flag（できないとき１、それ以外のとき０となるものとする。）とチェックフラグ生成部５４が生成したチェックフラグfullcheckとの論理積をとる（Ｓ１４）。そしてこの結果OFIFO_fullが１か否かを判定する（Ｓ１５）。結果OFIFO_fullが１のときには、処理はＳ１４に戻る（Ｓ１５にて「ＹＥＳ」）。その結果、出力ＦＩＦＯ４４のデータが読出されてOFIFO_full_flagがリセットされるまでＳ１４およびＳ１５の処理が繰返され、出

力ＦＩＦＯ４４への次のデータの書込みをする命令の処理が止められる。なおOFIFO_full_flagは、Ｎ個のＦＩＦＯのバンクのうち、Ｎ－１個まではデータが書込まれており、全てのデータがまだ読出されていない状態で、Ｎ番目のバンクにデータが初めて書込まれたときにセットされるものとする。

【００３０】Ｓ１５の判定の結果が「ＮＯ」となると処理はＳ１１に戻り、次の命令に対する処理が行なわれる。

【００３１】このような構成とすることにより、出力ＦＩＦＯ４４に書込まれるデータの数が、データを出力ＦＩＦＯ４４に書込む前にカウントされ、バンクの終わりには必ず出力ＦＩＦＯ４４の空き領域の大きさが所定の条件を充足しているか否かが調べられる。その結果、バンクの境界をまたがるようなデータ転送があった場合に、、ハードウェアにより高速にバンクがＦｕｌｌとなるおそれがあるか否かを判定することができプログラム側でバンクがＦｕｌｌか否かを判定する必要がなくなる。処理速度の遅いソフトウェア処理が避けられ、データ転送速度を向上させることができる。

【００３２】この実施の形態１の出力ＦＩＦＯデータ転送制御装置の動作を具体的な例をあげて説明する。

【００３３】図３に示すように出力ＦＩＦＯ４４が４つのバンクBank１～４を含み、各バンクのサイズが３２であるものとする。この時、式（２）に示すＮＮ＝Ｎ－Ｍの値はＮＮ＝３２－４＝２８である。したがってCounter>=28となるとfullcheckがセットされる。

【００３４】図３において斜線で示した部分にはデータが書込済み（未読出）であるものとする。すなわち、バンク１、３、４は全て書込み済みであり、バンク２のみが２５個分だけ書込み済みであり未読出であるものとする。演算器の数は４つで、１命令で最大４個のデータが同時に転送される可能性があるものとする。書込中のバンクはバンク２である。このとき、次のバンク（バンク３）への書込はできないので、OFIFO_full_flagはセットされている。

【００３５】この状態で以下のような命令が順に発行されるものとし、そのときの出力ＦＩＦＯ４４の状態について説明する。

【００３６】
命令Ａ：１個のデータ転送
命令Ｂ：３個のデータ転送
命令Ｃ：４個のデータ転送
（１）（Ｓ１１：図２）
Ｆｕｌｌチェック部５６が命令Ａを解析し、ｍ＝１を求める。出力ＦＩＦＯ４４にデータを１個書込む（図４）。

【００３７】（２）（Ｓ１２）
データカウント部５２においてCounter=Counter+1が計算される。したがってCounter=25+1=26が求められる。

【００３８】（３）（Ｓ１３）
チェックフラグ生成部５４において、Countet>=28が成り立たないので、fullcheckはセットされない。

【００３９】（４）（Ｓ１４、１５）
fullcheckが０であるので、チェックフラグ生成部５４における判定の結果、処理はＳ１１に戻る。

【００４０】（５）（Ｓ１１）
命令解析部５０において、次の命令（命令Ｂ）を解析し、ｍ＝３を求める。さらに出力ＦＩＦＯ４４に対してデータを３個書込む（図５）。

【００４１】（６）（Ｓ１２）
データカウント部５２においてCounter=26+3=29を求める。

【００４２】（７）（Ｓ１３）
チェックフラグ生成部５４において、Counter>=28が成り立つのでfullcheckがセットされる。

【００４３】（８）（Ｓ１４）
Ｆｕｌｌチェック部５６において、fullcheckが１、OFIFO_full_flagも１であるのでOFIFO_fullは１となる。

【００４４】（９）（Ｓ１５）
OFIFO_fullが１であるので、OFIFO_full_flagがリセットされるまで（すなわち出力ＦＩＦＯ４４のバンク３のデータが読み出されるまで）、Ｆｕｌｌチェック部５６の処理（Ｓ１４およびＳ１５）を繰返す。バンク３のデータが読み出され、OFIFO_full_flagがリセットされるとＦｕｌｌチェック部５６での判定の結果処理はＳ１１に戻る。このときの出力ＦＩＦＯ４４の状態を図６に示す。

【００４５】（１０）（Ｓ１１）
命令解析部５０が次の命令Ｃを解析し、ｍ＝４を求め、出力ＦＩＦＯ４４に対して４個のデータを書込む。

【００４６】（１１）（Ｓ１２）
データカウント部５２においてCounter=29+4=33が求められる。

【００４７】（１２）（Ｓ１３）
チェックフラグ生成部５４は、Counter>=28が成り立つのでfullcheckをセットする。

【００４８】（１３）（Ｓ１４）
OFIFO_full_flagは０である。したがってＦｕｌｌチェック部５６の判定結果であるOFIFO_fullも０となる。

【００４９】（１４）（Ｓ１５）
OFIFO_fullが０であるためＳ１１に戻る（図７）。この後の命令でデータ転送が行なわれる場合には、バンク３のデータは既に読出済みである。したがってデータの上書きが生ずるおそれはない。

【００５０】以上のようにこの実施の形態の出力ＦＩＦＯデータ転送制御装置４２によれば、出力ＦＩＦＯ４４にデータを書込む際に、そのデータ数を求め、バンクの終わりでそのバンクの空き領域の大きさを調べるようにした。そして、バンクをまたがってデータを書込む可能

性がある場合には、出力ＦＩＦＯ４４に対するデータ転送が確実に行える状態となるまで次の命令を受付けない。そのため、バンクをまたがるデータの書込みが発生してもオーバーフローを回避でき、読み出されていないデータに上書きされるおそれはない。したがって、プログラム側でそうしたデータの上書きを防止するための方策をとる必要はない。その結果、データ転送処理速度の向上を図ることができる。

【００５１】実施の形態２

上に説明した実施の形態１では、出力ＦＩＦＯ４４が複数個のバンクからなっており、そのために上記した構成でうまく転送を行なうことができる。ところが、出力ＦＩＦＯがバンクを１個しか持たないときに実施の形態１と同様の方法を適用しようとすると、このバンクに実際には空き領域がある場合であっても一部でもデータが書込まれているとOFIFO_full_flagはリセットされない。その間、データの書込はストップされる。そのためにデータの転送処理速度が低下してしまうおそれがある。

【００５２】この実施の形態２において説明する出力ＦＩＦＯデータ転送制御装置は、出力ＦＩＦＯのバンクが１つの場合でも高速にデータ転送を行なうよう改良したものである。そのためにこの実施の形態２では、単にＦＩＦＯが全く空かそうでないかを示すフラグのみでなく、ＦＩＦＯにどの程度の空き領域があるかを調べ、空き領域がある場合にはOFIFO_full_flagをリセットし、ない場合にはセットする。以下この実施の形態２にかかる出力ＦＩＦＯデータ転送制御装置について詳細に説明する。

【００５３】図８を参照して、この実施の形態２にかかる出力ＦＩＦＯデータ転送制御装置７２を含むプロセッサ７０は、演算器４０と、出力ＦＩＦＯデータ転送制御装置７２と、一つのバンクからなり、出力ＦＩＦＯデータ転送制御装置７２からのデータを出力装置４６に出力するための出力ＦＩＦＯ７４とを含む。

【００５４】出力ＦＩＦＯデータ転送制御装置７２は、出力ＦＩＦＯ７４への書込み命令を解析するための命令解析部８０と、出力ＦＩＦＯ７４に空き領域があるか否かを調べるためのデータカウント部８２と、出力ＦＩＦＯ７４がＦｕｌｌ状態か否かを調べるためのＦｕｌｌチェック部８６とを含む。

【００５５】図９を参照して、出力ＦＩＦＯデータ転送制御装置７２は以下のように動作する。命令解析部８０は、演算器４０から出力ＦＩＦＯ７４へのデータの書込み命令を受けると、書込まれるデータ数（ｍ）を求め、出力ＦＩＦＯ７４にデータを書込む（Ｓ２１）。データカウント部８２は、自分が維持している出力ＦＩＦＯ７４内に格納されているデータ数を表わすカウンタの値ｎｕｍに、命令解析部８０が求めた値ｍを加算する。すなわち以下の計算が行なわれる（Ｓ２２）。

【００５６】ｎｕｍ＝ｎｕｍ＋ｍ　…(4)

データカウント部８２はさらに、出力ＦＩＦＯ７４からデータが読み出された場合には、読み出された数（ｒとする。）だけ、カウンタの値ｎｕｍから減算する。すなわち以下の計算が行なわれる（Ｓ２３）。

【００５７】ｎｕｍ＝ｎｕｍ-ｒ　…(5)

データカウント部８２はこうして計算して得られたカウンタの値ｎｕｍの値が出力ＦＩＦＯ７４のサイズ（Ｎ）から１命令で転送できるデータの最大数（Ｍ）を減算した値（Ｎ－Ｍ）以上か否かを判定する（Ｓ２４）。ｎｕｍ＞N-Mであればフラグ0FIFO_full_flag_mをセットし、それ以外であれば0FIFO_full_flag_mをリセットする（Ｓ２４）。これをＣ言語の記法にしたがって表現すると以下のとおりとなる。

【００５８】

0FIFO_full_flag_m=(num>=N-M) ? 1:0　…(6)

そしてＦｕｌｌチェック部８６は、0FIFO_full_flag_mが１であればＳ２３に戻り（Ｓ２５）、0FIFO_full_flag_mがリセットされるまでＳ２３～Ｓ２５の処理を繰返す。すなわち出力ＦＩＦＯデータ転送制御装置７２は、出力ＦＩＦＯ７４からデータが読み出されて式（６）内の条件式が成立しなくなるまで、次の命令を受付けない。式（６）内の条件式が成立すると制御はＳ２１に戻り、次の命令の処理を行なう。

【００５９】この出力ＦＩＦＯデータ転送制御装置７２の動作を具体的な例で説明する。図１０に示すように、出力ＦＩＦＯ７４はサイズが３２で１個のバンクを持つものとする。演算器４０の数は４つ、したがって最大４個のデータが同時に転送される可能性があるものとする。このとき、式（６）のNN＝N-MはNN＝32-4＝8となる。すなわち出力ＦＩＦＯデータ転送制御装置７２は、ｎｕｍ＞＝28のときに0FIFO_full_flag_mをセットする。

【００６０】たとえば、図１０に示すように出力ＦＩＦＯ７４に２５個のデータが書込まれており、１個も読み出されていない場合を想定する（ｎｕｍ＝25）。このとき、以下のような命令が発行され、かつ命令Ａと命令Ｂとの間に２個のデータが読み出されるものとする。

【００６１】

命令Ａ：１個転送（２個読出）

命令Ｂ：３個転送

命令Ｃ：４個転送

このとき、具体的な動作にしたがって出力ＦＩＦＯ７４の状態は以下のように変化する。

【００６２】（１）　（Ｓ２１）

命令解析部８０において命令Ａが解析され、ｍ＝１が求められる。また出力ＦＩＦＯ７４にデータが１個書込まれる。

【００６３】（２）（Ｓ２２）

データカウント部８２においてｎｕｍ＝25＋1＝26を求める（図１１）。

【００６４】（３）（Ｓ２３）
データが２個読み出されたことに応答して、データカウント部８２においてｎｕｍ＝26-2＝24が計算される（図１２）。
【００６５】（４）（Ｓ２４）
ｎｕｍ＞＝28が満足されないので、Ｆｕｌｌチェック部８６はOFIFO_full_flag_mをリセットする。
【００６６】（５）（Ｓ２５）
OFIFO_full_flag_mが０であるので、制御はＳ２１に戻り次の命令（命令Ｂ）の処理が開始される。
【００６７】（６）（Ｓ２１）
命令解析部８０が命令Ｂを解析しｍ＝３を求める。出力ＦＩＦＯ７４にデータが３個書込まれる。
【００６８】（７）（Ｓ２２、Ｓ２３）
データカウント部８２においてｎｕｍ＝24＋3＝27が求められる。データの読出はなく、ｒ＝0である。したがってｎｕｍ＝27である（図１３）。
【００６９】（８）（Ｓ２４）
ｎｕｍ＞＝28を満足しないので、Ｆｕｌｌチェック部８６はOFIFO_full_flag_mをリセットする。
【００７０】（９）（Ｓ２５）
OFIFO_full_flag_mが０であるので、制御はＳ２１に戻り、次の命令（命令Ｃ）の処理が開始される。
【００７１】（１０）（Ｓ２１）
命令解析部８０が命令Ｃを解析しｍ＝４を求める。出力ＦＩＦＯ７４にデータが４個書込まれる。
【００７２】（１１）（Ｓ２２、Ｓ２３）
データカウント部８２においてｎｕｍ＝ｎｕｍ＋ｍ＝27＋4＝31が求められる（図１４）。
【００７３】（１２）（Ｓ２４）
ｎｕｍ＞＝28が成り立つので、Ｆｕｌｌチェック部８６はOFIFO_full_flag_mをセットする。
【００７４】（１３）（Ｓ２５）
OFIFO_full_flag_mが１であるので、Ｆｕｌｌチェック部８６はOFIFO_full_flag_mが１になるまでＳ２３～Ｓ２５の処理を繰返す。出力ＦＩＦＯ７４からデータが４個以上読み出されてOFIFO_full_flag_mがリセットされ０になれば（図１５）、制御はＳ２１に戻りさらに次の命令の処理が開始される。
【００７５】以上のようにこの実施の形態の装置では、出力ＦＩＦＯ７４に空き領域があるか否かを調べ、十分な空き領域さえあれば全てのデータが読み出される前にデータを出力ＦＩＦＯ７４に書込むことができる。その結果、出力ＦＩＦＯ７４への書込みが停止するおそれが少なくなり、出力ＦＩＦＯ７４を効率良く使用してデータ転送処理速度を向上させることができる。
【００７６】実施の形態３
上に説明した実施の形態２では、出力ＦＩＦＯのバンク数が１であり、それ故に上記したように十分な空き領域があるか否かを調べることによってデータ転送処理速度の向上を図っている。しかし、出力ＦＩＦＯに十分な空き領域があるか否かを調べる手法は、出力ＦＩＦＯのバンク数が１のときだけに適用可能なわけではなく、バンク数が複数のときにも適用できる。以下に説明する実施の形態３は、そうした例である。
【００７７】図１６を参照して、この実施の形態３にかかる出力ＦＩＦＯデータ転送制御装置１１２を有するプロセッサ１００は、演算器４０と、出力ＦＩＦＯデータ転送制御装置１１２と、出力ＦＩＦＯデータ転送制御装置１１２から受けるデータを出力装置４６に出力する、複数個のバンクを含む出力ＦＩＦＯ４４とを含む。
【００７８】出力ＦＩＦＯデータ転送制御装置１１２は、演算器４０から受ける命令を解析するための命令解析部１２０と、出力ＦＩＦＯ４４に空き領域があるか否かを調べるためのデータカウント部１２２と、出力ＦＩＦＯ４４がＦｕｌｌ状態か否かを調べるためのＦｕｌｌチェック部１２６とを含む。
【００７９】図１７を参照して、この出力ＦＩＦＯデータ転送制御装置１１２は以下のように動作する。命令解析部１２０は、演算器４０から出力ＦＩＦＯ４４へのデータの書込み命令を受けると、書込まれるデータ数（ｍ）を求め、出力ＦＩＦＯ４４にデータを書込む（Ｓ２１）。データカウント部１２２は、自分が維持している出力ＦＩＦＯ４４内に格納されているデータ数を表わすカウンタの値ｎｕｍに、命令解析部１２０が求めた値ｍを加算する。すなわち以下の計算が行なわれる（Ｓ２２）。
【００８０】ｎｕｍ＝ｎｕｍ＋ｍ　…(4)
データカウント部１２２はさらに、出力ＦＩＦＯ４４からデータが読み出された場合には、読み出された数（ｒ）だけ、カウンタの値ｎｕｍから減算する。すなわち以下の計算が行なわれる（Ｓ２３）。
【００８１】ｎｕｍ＝ｎｕｍ－ｒ　…(5)
データカウント部１２２はこうして計算して得られたカウンタの値ｎｕｍの値が出力ＦＩＦＯ４４の各バンクのトータルサイズ（Ｎ＊Ｂ、ただしＢはバンク数）から１命令で転送できるデータの最大数（Ｍ）を減算した値（ＮＮ＝Ｎ＊Ｂ－Ｍ）以上か否かを判定する（Ｓ２４）。ｎｕｍ＞N*B-Mであればフラグ OFIFO_full_flag_mをセットし、それ以外であればOFIFO_full_flag_mをリセットする（Ｓ３４）。これをＣ言語の記法にしたがって表現すると以下のとおりとなる。
【００８２】
OFIFO_full_flag_m=(num>=N*B-M) ? 1:0　…(7)
そしてＦｕｌｌチェック部１２６は、OFIFO_full_flag_mが１であればＳ２３に戻り（Ｓ２５）、OFIFO_full_flag_mがリセットされるまでＳ２３～Ｓ２５の処理を繰返す。すなわち出力ＦＩＦＯデータ転送制御装置１１２は、出力ＦＩＦＯ４４からデータが読み出されて式（６）内の条件式が成立しなくなるまで、次の命令を受

付けない。式（６）内の条件式が成立すると制御はＳ２１に戻り、次の命令の処理を行なう。

【００８３】このときの出力ＦＩＦＯ４４の状態を具体的に説明する。図１８に示すように、出力ＦＩＦＯ４４が、各々サイズが３２の４つのバンクを有するものとする（Ｂ＝４）。演算器４０の数は４つで、したがって同時に４個のデータが出力ＦＩＦＯ４４に転送される可能性があるものとする（Ｍ＝４）。このとき、式（７）のＮＮ＝Ｎ＊Ｂ－Ｍ＝３２＊４－４＝１２４である。したがってｎｕｍ＞＝124のときにOFIFO_full_flag_mがセットされる。

【００８４】たとえば図１８に示すように出力ＦＩＦＯ４４に１２１個のデータが書込まれており、１個も読み出されていないものとする（ｎｕｍ＝121）。そして、以下のような命令が発行されるものとし、かつ命令Ａと命令Ｂとの間には出力ＦＩＦＯ４４から２個のデータが読み出されるものと仮定する。

【００８５】

命令Ａ：１個転送（２個読出）

命令Ｂ：３個転送

命令Ｃ：４個転送

このとき、具体的な動作にしたがって出力ＦＩＦＯ７４の状態は以下のように変化する。

【００８６】（１）　（Ｓ２１）

命令解析部１２０において命令Ａが解析され、ｍ＝１が求められる。また出力ＦＩＦＯ４４にデータが１個書込まれる。

【００８７】（２）（Ｓ２２）

データカウント部１２２においてｎｕｍ＝121＋1＝122を求める。

【００８８】（３）（Ｓ２３）

データが２個読み出されたことに応答して、データカウント部１２２においてｎｕｍ＝122-2＝120が計算される。

【００８９】（４）（Ｓ３４）

ｎｕｍ＞＝124が満足されないので、Ｆｕｌｌチェック部１２６はOFIFO_full_flag_mをリセットする。

【００９０】（５）（Ｓ２５）

OFIFO_full_flag_mが０であるので、制御はＳ２１に戻り次の命令（命令Ｂ）の処理が開始される。

【００９１】（６）（Ｓ２１）

命令解析部１２０が命令Ｂを解析しｍ＝３を求める。出力ＦＩＦＯ４４にデータが３個書込まれる。

【００９２】（７）（Ｓ２２、Ｓ２３）

データカウント部１２２においてｎｕｍ＝120＋3＝123が求められる。データの読出はなく、ｒ＝0である。したがってｎｕｍ＝123である。

【００９３】（８）（Ｓ２４）

ｎｕｍ＞＝124を満足しないので、Ｆｕｌｌチェック部１２６はOFIFO_full_flag_mをリセットする。

【００９４】（９）（Ｓ２５）

OFIFO_full_flag_mが０であるので、制御はＳ２１に戻り、次の命令（命令Ｃ）の処理が開始される。

【００９５】（１０）（Ｓ２１）

命令解析部１２０が命令Ｃを解析しｍ＝４を求める。出力ＦＩＦＯ４４にデータが４個書込まれる。

【００９６】（１１）（Ｓ２２、Ｓ２３）

データカウント部１２２においてｎｕｍ＝ｎｕｍ＋ｍ＝123＋4＝127が求められる。

【００９７】（１２）（Ｓ２４）

ｎｕｍ＞＝124が成り立つので、Ｆｕｌｌチェック部１２６はOFIFO_full_flag_mをセットする。

【００９８】（１３）（Ｓ２５）

OFIFO_full_flag_mが１であるので、Ｆｕｌｌチェック部１２６はOFIFO_full_flag_mが０になるまでＳ２３～Ｓ２５の処理を繰返す。出力ＦＩＦＯ４４からデータが４個以上読み出されてOFIFO_full_flag_mがリセットされ０になれば、制御はＳ２１に戻りさらに次の命令の処理が開始される。

【００９９】以上のようにしてこの実施の形態の装置においても、出力ＦＩＦＯ４４が複数個のバンクを有している場合にも、出力ＦＩＦＯ４４の空き領域をカウントすることにより出力ＦＩＦＯ４４にデータ転送のために十分な領域があるか否かを判定し、十分な領域があるのであれば、出力ＦＩＦＯ４４のデータが全て読み出されるのを待たずにデータを出力ＦＩＦＯ４４に書込むことができる。その結果、データ転送処理速度向上させることができる。

【０１００】実施の形態４

実施の形態１～３においては、出力ＦＩＦＯの各バンクに書込まれたデータの数をカウントして、出力ＦＩＦＯの空き領域の大きさが所定の条件を充足しているか否かを調べる。この空き領域を調べるための手段としては以下の方法もある。

【０１０１】図１９を参照して、この第４の実施の形態にかかるプロセッサ１３０は、図１に示す第１の実施の形態の出力ＦＩＦＯデータ転送制御装置４２に替えて、以下に述べる点で出力ＦＩＦＯデータ転送制御装置４２と異なる出力ＦＩＦＯデータ転送制御装置１３２を含む。出力ＦＩＦＯデータ転送制御装置１３２は、出力ＦＩＦＯデータ転送制御装置４２のデータカウント部５２に替えて、以下に述べるような方法にしたがって出力ＦＩＦＯ４４の空き領域の大きさを調べる空き領域算出部１４４を有する点のみで異なる。図１９において、図１と同じ部品には同じ参照符号を付してある。それらの機能および名称も同一である。したがってここではそれらについての詳細な説明は繰返さない。

【０１０２】図２０を参照して、空き領域算出部１４４の機能も含めて出力ＦＩＦＯデータ転送制御装置１３２の動作について説明する。まず命令解析部５０は、命令

を解析して次に出力ＦＩＦＯ４４に書込まれるべきデータの数（ｍ）を算出し、このデータを出力ＦＩＦＯ４４に書込む（Ｓ１１）。空き領域算出部は、出力ＦＩＦＯ４４の各バンクに書込むことができるデータの数を示すカウンタの値Counter（初期値はCounter=N）からｍの値を減算する。すなわち、以下の式によってカウンタの値Counterを計算する（Ｓ４２）。

【０１０３】Counter = Counter - m

チェックフラグ生成部５４は、カウンタの値Counterと、出力ＦＩＦＯ４４に１命令で書込めるデータの最大数（Ｍ）とを比較する。チェックフラグ生成部５４は、カウンタの値CounterがM未満の場合には、チェックフラグfullcheckをセットし、他の場合にはリセットする（Ｓ１３）。これをＣ言語の記法にしたがって書けば以下のとおりとなる。

【０１０４】fullcheck = (Counter < M) ? 1:0

Ｆｕｌｌチェック部５６は、出力ＦＩＦＯ４４の次のバンクへの書込みができるか否かを示すフラグOFIFO_full_flag（できないとき１、それ以外のとき０となるものとする。）とチェックフラグ生成部５４が生成したチェックフラグfullcheckとの論理積をとる（Ｓ１４）。そしてこの結果OFIFO_fullが１か否かを判定する（Ｓ１５）。

【０１０５】以後の処理は実施の形態１の場合と同様である。またこの実施の形態４によっても実施の形態１と同様の効果を得ることができる。

【０１０６】実施の形態５

実施の形態２についても、同様な変形を行なうことができる。すなわち、図８に示す実施の形態２では、出力ＦＩＦＯに書込まれているデータ数をカウントするためのデータカウント部８２を設け、その計算結果に基づいて出力ＦＩＦＯ７４がフルか否かをｆｕｌｌチェック部８６で判定している。これに替えて、書込まれたデータ数と読出されたデータ数とから、出力ＦＩＦＯ７４においてさらに書込が可能な領域の大きさを直接算出して維持するための手段を設け、その書込が可能な大きさと出力ＦＩＦＯ７４に１度に書込むことができる最大のデータ数とを比較するような構成にすることができる。図２１に示す実施の形態５の装置はそのような装置である。なおハードウェア的構成は図８に示したものとほぼ同様であるので、ここでは制御の流れを図２１のフローチャートにしたがって説明する。なお以下の説明では、各部の参照符号として図８に示したものを用いる。また、図８のデータカウント部８２にかえて空き領域算出部を持つものとする（図示せず）。

【０１０７】図２１を参照して、この出力ＦＩＦＯデータ転送制御装置は、以下のように動作する。命令解析部８０は、演算器４０から出力ＦＩＦＯ７４へのデータの書込み命令を受けると、書込まれるデータ数（ｍ）を求め、出力ＦＩＦＯ７４にデータを書込む（Ｓ２１）。空き領域算出部は、自分が維持している出力ＦＩＦＯ７４内に書込めるデータ数を表わすカウンタの値ｎｕｍから、命令解析部８０が求めた値ｍを減算する。すなわち以下の計算が行なわれる（Ｓ５２）。

【０１０８】ｎｕｍ＝ｎｕｍ－ｍ

空き領域算出部はさらに、出力ＦＩＦＯ７４からデータが読み出された場合には、読み出された数（ｒとする。）を、カウンタの値ｎｕｍに加算する。すなわち以下の計算が行なわれる（Ｓ５３）。

【０１０９】ｎｕｍ＝ｎｕｍ＋ｒ　…(5)

空き領域算出部はこうして計算して得られたカウンタの値ｎｕｍの値が、出力ＦＩＦＯ７４に１命令で転送できるデータの最大数（Ｍ）未満か否かを判定する（Ｓ２４）。ｎｕｍ＜ＭであればフラグOFIFO_full_flag_mをセットし、それ以外であればOFIFO_full_flag_mをリセットする（Ｓ５４）。これをＣ言語の記法にしたがって表現すると以下のとおりとなる。

【０１１０】

OFIFO_full_flag_m=(num<M) ? 1:0　…(6)

そしてＦｕｌｌチェック部８６は、OFIFO_full_flag_mが１であればＳ２３に戻り（Ｓ２５）、OFIFO_full_flag_mがリセットされるまでＳ２３～Ｓ２５の処理を繰返す。

【０１１１】以後の処理は実施の形態２の場合と同様であり、同様の効果を得ることができる。

【０１１２】同様の考えを実施の形態３についても適用することができる。ただしこの場合、ｎｕｍの初期値はＮ＊Ｂである。

【０１１３】今回開示された実施の形態はすべての点で例示であって制限的なものではないと考えられるべきである。本発明の範囲は上記した説明ではなくて特許請求の範囲によって示され、特許請求の範囲と均等の意味および範囲内でのすべての変更が含まれることが意図される。

【０１１４】

【発明の効果】以上のようにこの発明によれば、バンクの最後まで到達する前に、次のデータを書込んでも不都合がないか否かが必ず判定されるため、データの上書きが生ずるおそれはない。プログラムでデータの上書きを防止する処理を行なう必要もなく、データ転送処理速度の向上を図ることができる。

【０１１５】一回に転送可能な最大データ量以上の空き容量が確保されたとき、または次のバンクがフルでなくなったときに次の命令の処理が開始されるようにした場合には、次の命令で最大量のデータが出力ＦＩＦＯ記憶装置に転送されても上書きされたりするおそれはないので、プログラムでデータの上書きを防止する処理を行なう必要もなく、データ転送処理速度の向上を図ることができる。

【０１１６】この発明の他の局面によれば、必ず十分な

空き容量があることが確認されてから次の命令が処理されるため、データの上書きが生ずるおそれはない。またプログラムでデータの上書きを防止する処理を行なう必要もなく、データ転送処理速度の高速化を図ることができる。

【０１１７】出力ＦＩＦＯ記憶装置が複数個のバンクを含んでいる場合でも、単一のバンクのみ含んでいる場合でも、データ転送処理速度を高速化することができる。

【０１１８】出力ＦＩＦＯ記憶装置の空き容量が１回に転送可能な最大データ量以上のときのみ次の命令が処理されるようにすると、次の命令で転送されるデータ量が最大データ量であったとしてもデータの上書きが生ずるおそれはない。そのためプログラムにおいてデータの上書きを回避するための処理を行なうことが不要で、データ転送処理速度の高速化を図ることができる。

【図面の簡単な説明】

【図１】　本発明の実施の形態１にかかる出力ＦＩＦＯデータ転送制御装置を含むプロセッサのブロック図である。

【図２】　本発明の実施の形態１にかかる出力ＦＩＦＯデータ転送制御装置の制御プログラムのフローチャートである。

【図３】　本発明の実施の形態１にかかる出力ＦＩＦＯデータ転送制御装置の動作を示すためのＦＩＦＯの模式図である。

【図４】　本発明の実施の形態１にかかる出力ＦＩＦＯデータ転送制御装置の動作を示すためのＦＩＦＯのバンク２の模式図である。

【図５】　本発明の実施の形態１にかかる出力ＦＩＦＯデータ転送制御装置の動作を示すためのＦＩＦＯのバンク２の模式図である。

【図６】　本発明の実施の形態１にかかる出力ＦＩＦＯデータ転送制御装置の動作を示すためのＦＩＦＯのバンク２の模式図である。

【図７】　本発明の実施の形態１にかかる出力ＦＩＦＯデータ転送制御装置の動作を示すためのＦＩＦＯのバンク２の模式図である。

【図８】　本発明の実施の形態２にかかる出力ＦＩＦＯデータ転送制御装置を含むプロセッサのブロック図である。

【図９】　本発明の実施の形態２にかかる出力ＦＩＦＯデータ転送制御装置の制御プログラムのフローチャートである。

【図１０】　本発明の実施の形態２にかかる出力ＦＩＦＯデータ転送制御装置の動作を示すためのＦＩＦＯのバンク２の模式図である。

【図１１】　本発明の実施の形態２にかかる出力ＦＩＦＯデータ転送制御装置の動作を示すためのＦＩＦＯのバンク２の模式図である。

【図１２】　本発明の実施の形態２にかかる出力ＦＩＦＯデータ転送制御装置の動作を示すためのＦＩＦＯのバンク２の模式図である。

【図１３】　本発明の実施の形態２にかかる出力ＦＩＦＯデータ転送制御装置の動作を示すためのＦＩＦＯのバンク２の模式図である。

【図１４】　本発明の実施の形態２にかかる出力ＦＩＦＯデータ転送制御装置の動作を示すためのＦＩＦＯのバンク２の模式図である。

【図１５】　本発明の実施の形態２にかかる出力ＦＩＦＯデータ転送制御装置の動作を示すためのＦＩＦＯのバンク２の模式図である。

【図１６】　本発明の実施の形態３にかかる出力ＦＩＦＯデータ転送制御装置を含むプロセッサのブロック図である。

【図１７】　本発明の実施の形態３にかかる出力ＦＩＦＯデータ転送制御装置の制御プログラムのフローチャートである。

【図１８】　本発明の実施の形態３にかかる出力ＦＩＦＯデータ転送制御装置の動作を示すためのＦＩＦＯの模式図である。

【図１９】　本発明の実施の形態４にかかる出力ＦＩＦＯデータ転送制御装置を含むプロセッサのブロック図である。

【図２０】　本発明の実施の形態４にかかる出力ＦＩＦＯデータ転送制御装置の制御プログラムのフローチャートである。

【図２１】　本発明の実施の形態５にかかる出力ＦＩＦＯデータ転送制御装置の制御プログラムのフローチャートである。

【図２２】　従来の技術にかかる出力ＦＩＦＯデータ転送制御装置を含むプロセッサのブロック図である。

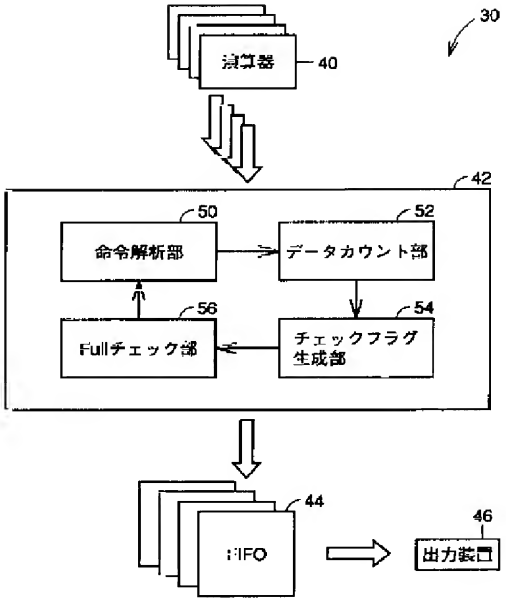
【図２３】　従来の技術にかかる出力ＦＩＦＯデータ転送制御装置の制御プログラムのフローチャートである。

【図２４】　従来の技術にかかる出力ＦＩＦＯデータ転送制御装置の動作を示すためのＦＩＦＯの模式図である。

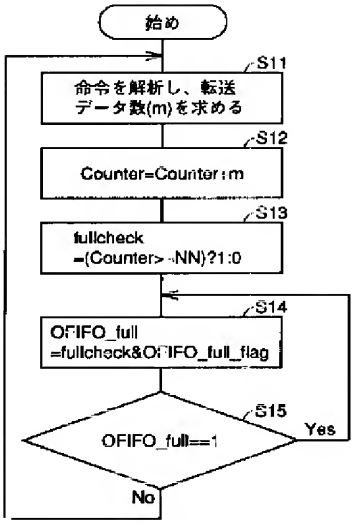
【符号の説明】

３０，７０，１００，１３０，１４０　プロセッサ、４０　演算器、４２，７２，１１２，１３２，１４２　出力ＦＩＦＯデータ転送制御装置、５０，８０，１２０，１５０　命令解析部、５２，８２，１２２，１５２　データカウント部、５４，１５４　チェックフラグ生成部、５６，８６，１２６，１５６　Ｆｕｌｌチェック部、１３４　空き領域算出部。

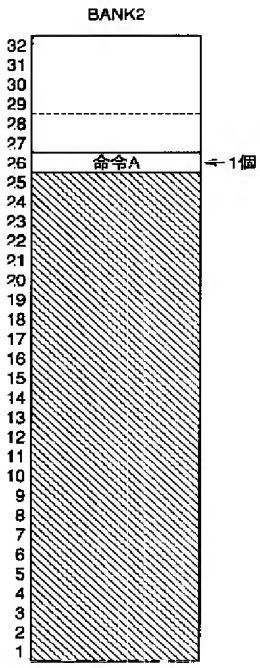
【図１】



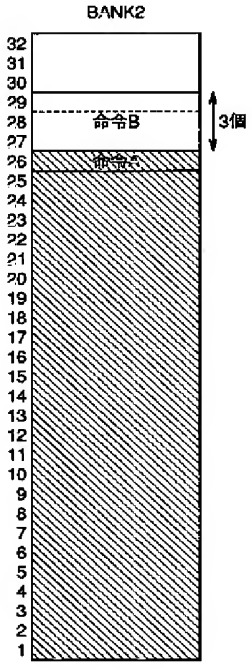
【図２】



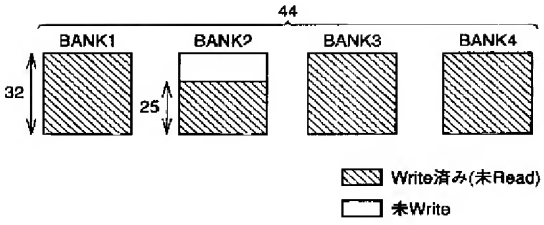
【図４】



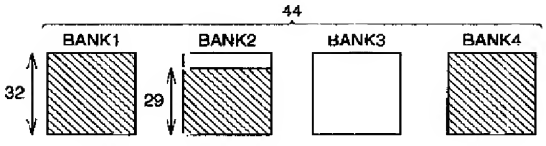
【図５】



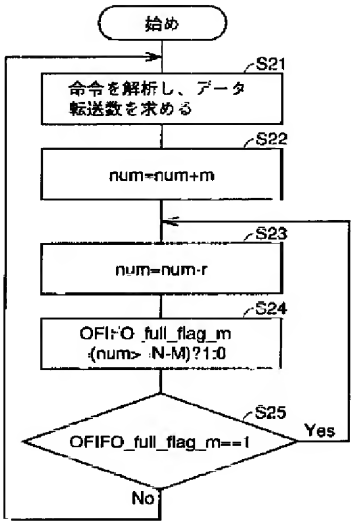
【図３】



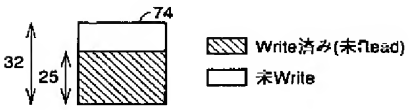
【図６】



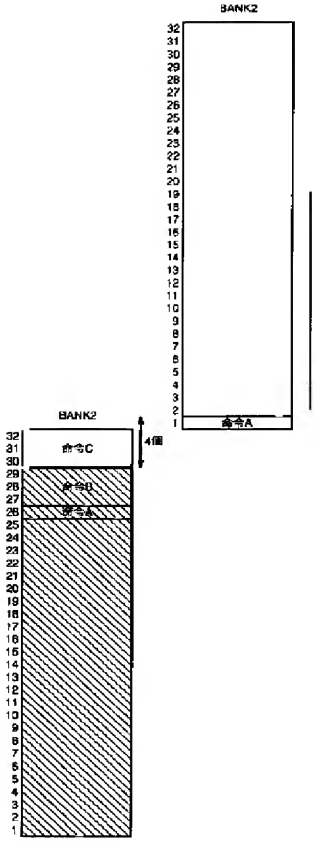
【図９】



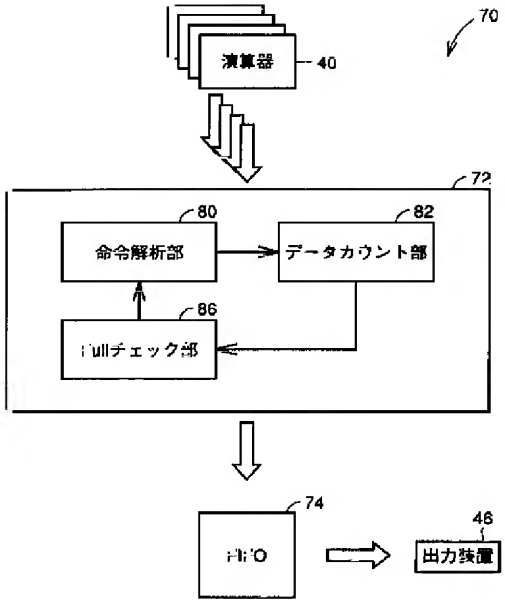
【図１０】



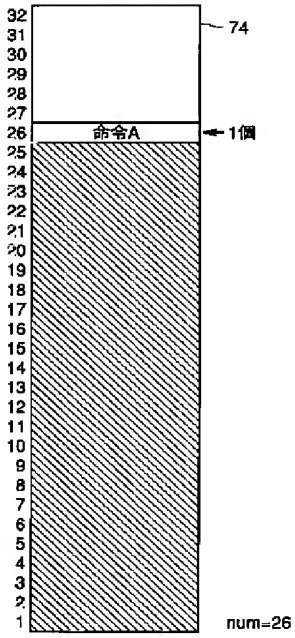
【図7】



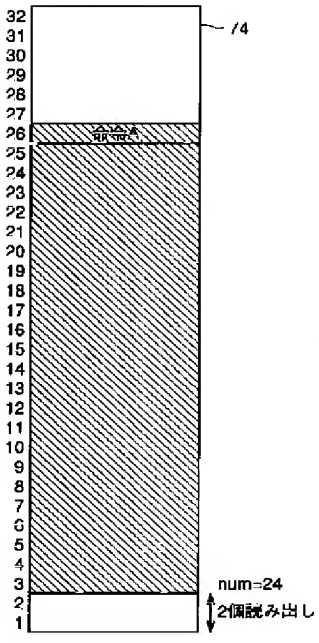
【図8】



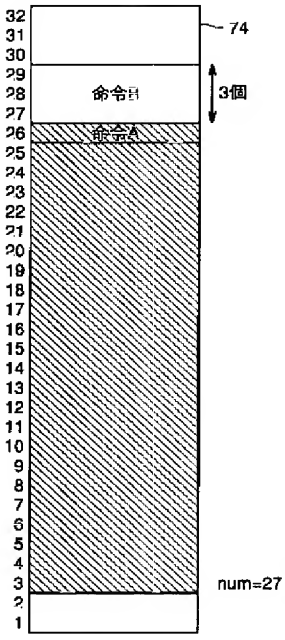
【図11】



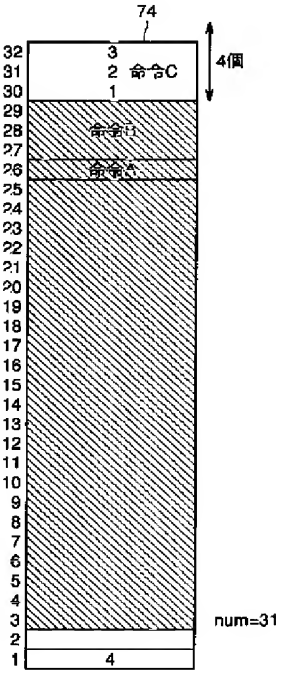
【図12】



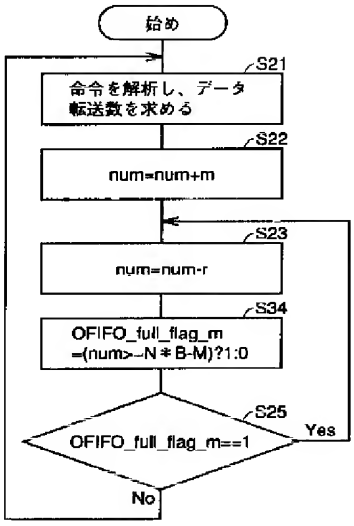
【図13】



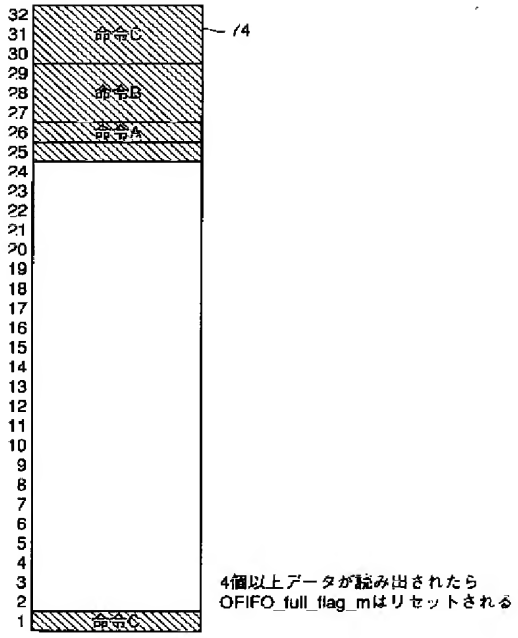
【図14】



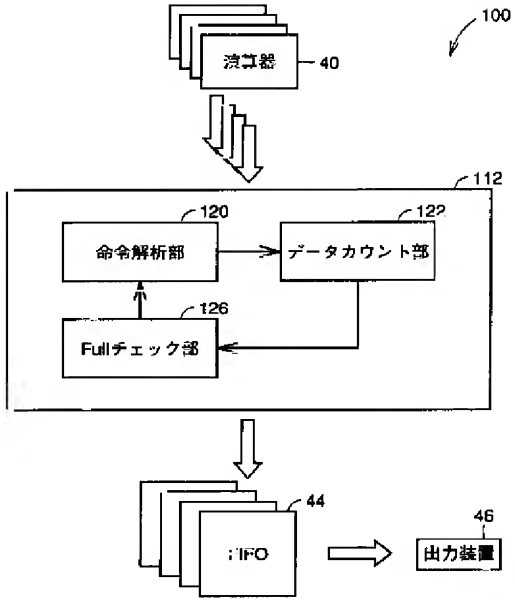
【図17】



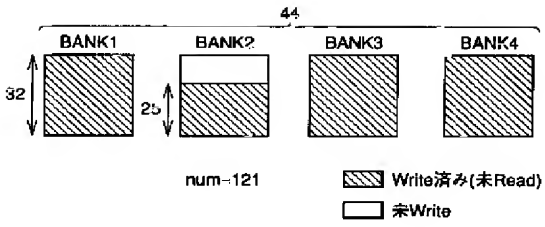
【図15】



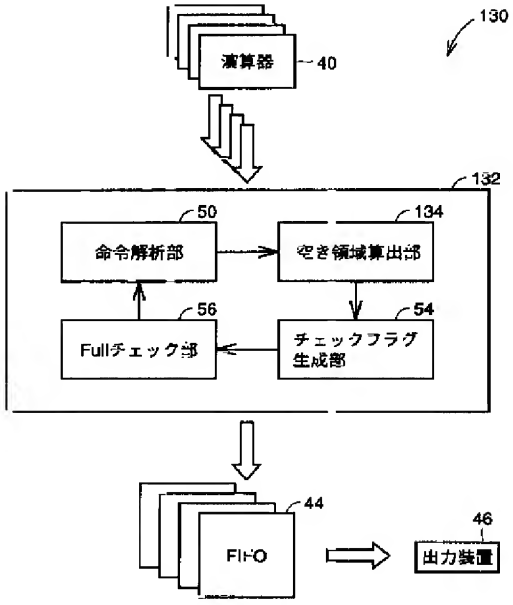
【図16】



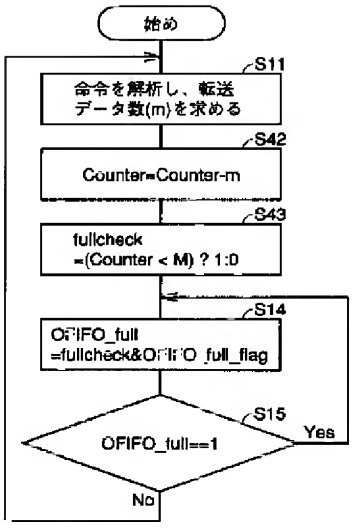
【図18】



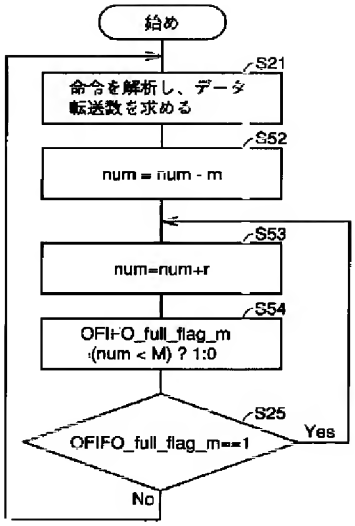
【図19】



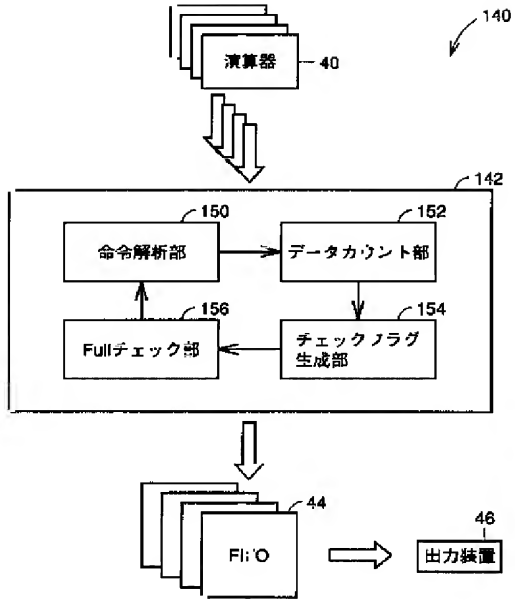
【図20】



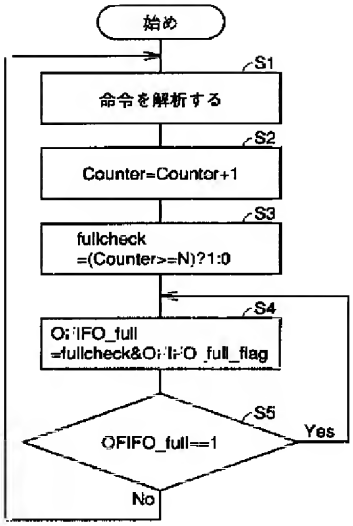
【図２１】



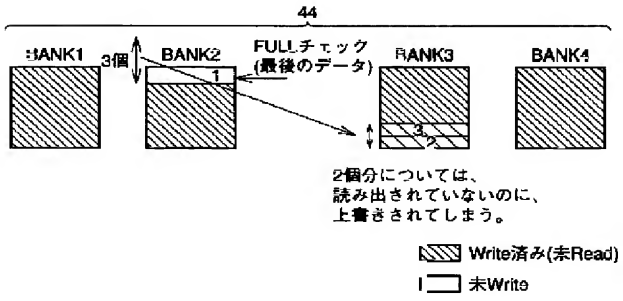
【図２２】



【図２３】



【図２４】



フロントページの続き

(51)Int.Cl.⁷	識別記号	FI	(参考)
G06F 13/38	310	G06F 13/38	310D

(72)発明者　井上　喜嗣
東京都千代田区丸の内二丁目２番３号　三菱電機株式会社内

(72)発明者　ロバート・ストライテンベルガー
東京都千代田区丸の内二丁目２番３号　三菱電機株式会社内

Fターム(参考)　5B060 AA14 AC11 CA17
5B077 AA15 AA45 BA02 DD01 DD12
DD17 DD22 MM03